Complexity Theory

Organised by
Joachim von zur Gathen, Universität Bonn
Oded Goldreich, Weizmann Institute, Rehovot
Madhu Sudan, MIT, Cambridge

June 24th – June 30th, 2007

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. Connections to the theory of error-correcting codes played a central role in many of these developments.

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

Introduction by the Organisers

The workshop Complexity Theory was organized by Joachim von zur Gathen (Universität Bonn), Oded Goldreich (Weizmann Institute), and Madhu Sudan (MIT). The workshop was held on June 24th–30th 2007, 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 eight long lectures as well as short (10-minute) reports by almost all participants. In addition, extensive 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 number theory, algebra, combinatorics, coding theory, and optimization.

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.

**Connections to the Theory of Error-Correcting Codes.** The interplay between coding theory and complexity theory first emerged in the context of “hardness amplification” (almost two decades ago) and other connections are less than a decade old (e.g., the connection to probabilistic checking of proofs and extraction of pure randomness). Several applications of the known connections were presented in the current meeting, and in addition a new connection to algebraic complexity was presented.

While previous applications of the aforementioned connections went in the direction of coding theory to complexity theory, a recent result reported by Venkat Guruswami goes in the opposite direction. This work, by Guruswami and his graduate student (Rudra), resolves a decades-old central problem in coding theory by presenting an explicit error-correcting code of constant-size alphabet that approaches the capacity bound (under worst-case errors, using list decoding).

**Extracting randomness.** Extracting almost-perfect randomness from weak sources of (imperfect) randomness is crucial for the actual use of randomized procedures. Typical analyses of randomized procedures assume that the procedures have access to a perfect random source. However, in reality one only has access to sources of weak randomness (e.g., having constant entropy rate). Indeed, the problem has attracted a lot of attention in the last couple of decades.

In the meeting, Chris Umans has presented recent work with Guruswami and Vadhan, which utilizes recent algebraic and coding theoretic techniques to the construction of (single-source) randomness extractors. This construction meets (and actually improves) the best known parameters for the problem (which are almost optimal), but does so by a relatively simple construction rather than by a complex combination of numerous constructs (as done in prior work). Furthermore, the new work introduces improved constructions for an intermediate primitive (called randomness condenser), which is of independent interest.

While single-source randomness extractors must utilize an auxiliary random seed (which may be very short), some applications do not allow for such a seed. In this case, extraction from several (e.g., two) independent sources of weak randomness is called for. An important step in the study of this direction was made by Anup Rao, and presented by him in the meeting.
**Algebraic complexity and modular polynomial composition.** An important task in algebraic computation is modular polynomial composition; that is, given three univariate polynomials $f, g$ and $h$, one is required to obtain the coefficients of the polynomial $f \circ g \mod h$. This task has many applications, most notably as an ingredient in algorithms for polynomial factorization. The previously best algorithm was presented 30 years ago and uses $O(n^{1.7})$ arithmetic operations, where $n$ denotes the maximum degree of the polynomials.

In the meeting, Chris Umans presented significant progress on this celebrated open problem in the form of an almost linear-time algorithm that works for fields of small characteristic. This major progress on a purely algebraic problem is essentially based on methods that were introduced into coding theory by Guruswami and Rudra, and then applied to complexity theory in the context of randomness extractors (see foregoing paragraphs). All three results, which are major achievements in their respective areas, were presented at the meeting.

**Cryptography and Zero-Knowledge.** Zero-knowledge proofs are fascinating concepts and extremely useful constructs. Their fascinating nature is due to their seemingly contradictory definition that mandates that they be convincing and yet yield nothing beyond the validity of the assertion being proved. Their applicability in the domain of cryptography is vast; they are typically used to force malicious parties to behave according to a predetermined protocol. In addition to their direct applicability in cryptography, zero-knowledge proofs serve as a good benchmark for the study of various problems regarding cryptographic protocols. Zero-knowledge proofs come in many flavors, and it is of great theoretical and practical importance to investigate the relationship among them.

A central problem in this area, which has been open since 1986, refers to the gap between the known results regarding two dual notions: the notion of general zero-knowledge proofs (in which the secrecy condition holds with respect to feasible adversaries) and the notion of statistical zero-knowledge arguments (in which the soundness condition holds with respect to feasible adversaries). This gap was bridged in a recent work of Salil Vadhan, jointly with his graduate students (Nguyen and Ong), and was presented by Vadhan in this meeting.

A problem related to both cryptography and coding theory is the problem of constructing private information retrieval schemes and/or locally decodable codes. In the context of error-correcting codes, such schemes should allow the recovery of any bit in the original message based on a constant number (e.g., three) probes to the corrupted codeword. For more than a decade it was believed that the length of such codewords must be (weakly) exponential in the length of the message. In the meeting, Sergey Yekhanin (PhD student) presented his recent result that refutes this belief.

**Delegating your work to an untrusted entities.** Needless to say, it is nice to delegate your work to others, but what if you don't trust the others? The very definition of a proof system refers to such a possibility - the hard task of finding a proof is delegated to the outside while you make sure that the proof is valid by performing the easier task of verification. However, facilitating verification may
mean making the task of finding adequate proofs even harder. In the context of program checking this phenomenon is explicitly disallowed: wishing to solve some problem you may use an untrusted program that supposedly solves this problem (but not a program that solve more complex problems). Needless to say, the aim is allowing the delegator, called a checker, to use significantly fewer resources than any program that correctly solves the problem. In the meeting, Shafi Goldwasser presented a recent result that achieves this goal for a natural complexity measure (circuit depth) and for a wide class of problems (i.e., NC).

A characterization of testable graph properties. The area of property testing is concerned with promise problems that call for distinguishing those objects that have a predetermined property from objects that are “far” from any object having this property. The focus is on sub-linear time algorithms that probe the given object at few (randomly selected) locations. In some cases, one may perform the task by using a number of probes that only depends on the proximity parameter (and is independent of the size of the object). In the meeting, Noga Alon presented a recent result that characterizes the class of graph properties (where graphs are represented by their adjacency matrices) for which such a phenomenon holds.

The rest of this report. This report contains extended abstracts of the eight plenary lectures as well as abstracts of forty short reports.
Workshop: Complexity Theory

Table of Contents

Noga Alon (joint with Eldar Fischer, Ilan Newman, Asaf Shapira)
Characterizing the testable graph properties via the regularity lemma . . . . . . . 9

Shafi Goldwasser (joint with Dan Gutfreund, Alexander Healy, Tali
Kaufman), and Guy Rothblum)
A (De)constructive Approach to Program Checking (or How to Delegate
Work to Your Very Own Adversary) ............................................. 11

Venkatesan Guruswami (joint with Atri Rudra)
Reed-Solomon codes, low-degree automorphisms, and Error-correction
with optimal rate ................................................................. 14

Anup Rao
Extractors for Independent Sources .............................................. 17

Chris Umans (joint with Venkatesan Guruswami, Salil Vadhan)
Unbalanced Expanders and Randomness Extractors from Parvaresh-Vardy
Codes .................................................................................. 20

Salil Vadhan (joint with I. Haitner, M. Nguyen, S.J. Ong, O. Reingold)
Statistically Hiding Commitments from Any One-Way Function .............. 23

Avi Wigderson (joint with Emanuele Viola)
Norms, XOR lemmas, and lower bounds for GF(2) polynomials and
multiparty protocols .............................................................. 26

Sergey Yekhanin
Towards 3-Query Locally Decodable Codes of Subexponential Length ...... 29

Benny Applebaum (joint with Yuval Ishai, Eyal Kushilevitz)
Cryptography with Constant Input Locality ..................................... 32

Boaz Barak (joint with Mohammad Mahmoody-Ghidary)
Lower Bounds on Signatures From Symmetric Primitives ..................... 33

Paul Beame (joint with T. S. Jayram, Atri Rudra)
Lower Bounds for Randomized Read/Write Stream Algorithms .............. 33

Markus Bläser (joint with Holger Dell, Mahmoud Fouz, Christian
Hoffmann, Johann A. Makowskiy)
On the complexity of graph polynomials .................................... 34

Johannes Blömer (joint with Stefanie Naewe)
Sampling methods for shortest vectors, closest vectors and successive
minima .................................................................................. 35
Andrej Bogdanov (joint with Emanuele Viola)

Pseudorandom generators for low degree polynomials ................. 35

Peter Bürgisser

On defining integers and proving arithmetic circuit lower bounds .... 36

Claus Diem

On subexponentiality of the discrete logarithm problem in elliptic curves over extension fields .................................................. 37

Zeev Dvir (joint with Ariel Gabizon, Avi Wigderson)

Extractors and Rank Extractors for Polynomial Sources ............... 38

Joachim von zur Gathen

Counting reducible and singular bivariate polynomials ................. 39

Johan Håstad

On the Approximation Resistance of a Random Predicate ............... 40

Valentine Kabanets (joint with Ali Juma, Charles Rackoff, Amir Shpilka)

The Black-Box Query Complexity of Polynomial Summation .......... 41

Erich Kaltofen (joint with Zhengfeng Yang, Lihong Zhi)

On probabilistic analysis of randomization in hybrid symbolic-numeric algorithms ................................................................. 42

Guy Kindler (joint with Uri Feige, Ryan O'Donnell)

Understanding parallel repetition requires understanding foams ... 42

Pascal Koiran (joint with Sylvain Perifel)

Interpolation in Valiant's theory ............................................ 43

Swastik Kopparty (joint with Eli Ben-Sasson, Jaikumar Radhakrishnan)

Subspace polynomials and List Decoding of Reed-Solomon Codes .... 44

Matthias Krause (joint with Hellen Altendorf, Frederik Armknecht)

Constructing Boolean Functions of Maximal Algebraic Immunity .... 45

Noam Livne

All Natural NPC Problems Have Average-Case Complete Versions .... 46

Martin Lotz (joint with Peter Bürgisser, Felipe Cucker)

The probability that a small perturbation of a numerical analysis problem is difficult .......................................................... 47

Or Meir

Combinatorial Construction of Locally Testable Codes ................. 48

Dana Moshkovitz (joint with Ran Raz)

Sub-Constant Error Low Degree Test and PCP of Almost-Linear Size ... 48

Moni Naor (joint with Danny Harnik)

The Cryptographic Applications of Compressibility With Respect to Solutions ................................................................. 49
Ryan O'Donnell
Two problems related to the Max-Cut and the Unique Games Conjecture .......................... 50

Toniann Pitassi (joint with Konstantinos Georgiou, Avner Magen, Iannis Tourlakis)
Tight Integrality gaps for Vertex Cover SDPs in the Lovasz-Schrijver hierarchy .......................... 51

Pavel Pudlák
Quantum Frege proofs and a problem on quantum computing .................................................. 52

Ran Raz
Randomness versus Hardness and Lower Bounds for Constant-Depth Arithmetic Circuits .......................... 53

Alexander Razborov
Flag Algebras and Density of Triangles in Graphs ................................................................. 53

Oded Regev (joint with Avraham Ben-Aroya and Ronald de Wolf)
A Hypercontractive Inequality for Matrix-Valued Functions .................................................... 54

Omer Reingold (joint with Ronen Gradwohl)
Partial Exposure and Correlated Types in Large Games .......................................................... 55

Rüdiger Reischuk (joint with A. Jakoby, M. Liśkiewicz, C. Schindelhauer)
Designing Boolean Sorting Circuits with Optimal Average Delay ............................................ 56

Nitin Saxena
Some Specific Derandomizations .................................
David Zuckerman (joint with Xin Li, Anup Rao)

*Network Extractor Protocols and Three-Source Extractors* 65
Abstracts

Characterizing the testable graph properties via the regularity lemma

Noga Alon

(joint work with Eldar Fischer, Ilan Newman, Asaf Shapira)

A graph $G$ on $n$ vertices is $\epsilon$-far from satisfying a property $P$, if one needs to add and/or delete at least $n\epsilon^2$ edges to $G$ in order to turn it into a graph satisfying $P$. A tester for $P$ should distinguish with high probability, say $2/3$, between the case that $G$ satisfies $P$ and the case that $G$ is $\epsilon$-far from satisfying $P$. Here we assume that the tester can query some oracle whether a pair of vertices, $i$ and $j$, are adjacent in the input graph $G$.

Definition 1 (Testable). A graph property $P$ is testable if there is a randomized algorithm $T$, that can distinguish for every $\epsilon > 0$ and with probability $2/3$, between graphs satisfying $P$ and graphs that are $\epsilon$-far from satisfying $P$, while making a number of edge queries which is bounded by some function $q(\epsilon)$ that is independent of the size of the input.

This notion was introduced by Goldreich, Goldwasser and Ron [2] and received a considerable amount of attention by various researchers. The problem of characterizing the testable graph properties was naturally one of the main open problems in the study of the area and was raised already in this paper and mentioned in several subsequent ones. Here we give such a characterization. It is based on the regularity lemma of Szemerédi [3].

For every two nonempty disjoint vertex sets $A$ and $B$ of a graph $G$, we define $e(A,B)$ to be the number of edges of $G$ between $A$ and $B$. The edge density of the pair is defined by $d(A,B) = e(A,B)/(|A||B|)$.

Definition 2 ($\gamma$-regular pair). A pair $(A,B)$ is $\gamma$-regular, if for any two subsets $A' \subseteq A$ and $B' \subseteq B$, satisfying $|A'| \geq \gamma|A|$ and $|B'| \geq \gamma|B|$, the inequality $|d(A',B') - d(A,B)| \leq \gamma$ holds.

Definition 3 ($\gamma$-regular equipartition). An equipartition $B = \{V_i | 1 \leq i \leq k\}$ of the vertex set of a graph is called $\gamma$-regular if all but at most $\gamma \binom{k}{2}$ of the pairs $(V_i, V_j)$ are $\gamma$-regular.

An equipartition is said to refine another if every set of the former is contained in one of the sets of the latter. Szemerédi’s regularity lemma can be formulated as follows.

Lemma 4 ([3]). For every $m$ and $\gamma > 0$ there exists $T = T_4(m,\gamma)$ with the following property: If $G$ is a graph with $n \geq T$ vertices, and $A$ is any equipartition of the vertex set of $G$ of order at most $m$, then there exists a refinement $B$ of $A$ of order $k$, where $m \leq k \leq T$ and $B$ is $\gamma$-regular. In particular, for every $m$ and $\gamma > 0$ there exists $T = T_4(m,\gamma)$, such that any graph with $n \geq T$ vertices has a $\gamma$-regular equipartition of order $k$, where $m \leq k \leq T$. 
It seems natural to define a graph property, which states that a graph has a given $\gamma$-regular partition, that is, an equipartition which is $\gamma$-regular and such that the densities between the sets $V_i$ belong to some predefined set of densities.

**Definition 5 (Regularity-Instance).** A regularity-instance $R$ is given by an error-parameter $0 < \gamma \leq 1$, an integer $k$, a set of $\binom{k}{2}$ densities $0 \leq \eta_{ij} \leq 1$ indexed by $1 \leq i < j \leq k$, and a set $\overline{R}$ of pairs $(i, j)$ of size at most $\gamma \binom{k}{2}$. A graph is said to satisfy the regularity-instance if it has an equipartition $\{V_i\ 1 \leq i \leq k\}$ such that for all $(i,j) \notin \overline{R}$ the pair $(V_i, V_j)$ is $\gamma$-regular and satisfies $d(V_i, V_j) = \eta_{ij}$. The complexity of the regularity-instance is $\max(k, 1/\gamma)$.

The first main result in this work is the following:

**Theorem 6.** For any regularity-instance $R$, the property of satisfying $R$ is testable.

**Definition 7 (Regular-Reducible).** Graph property $P$ is regular-reducible if for any $\delta > 0$ there exists an $r = r_P(\delta)$ such that for any $n$ there is a family $R$ of at most $r$ regularity-instances each of complexity at most $r$, such that the following holds for every $\epsilon > 0$ and every $n$-vertex graph $G$:

1. If $G$ satisfies $P$ then for some $R \in R$, $G$ is $\delta$-close to satisfying $R$.
2. If $G$ is $\epsilon$-far from satisfying $P$, then for any $R \in R$, $G$ is $(\epsilon - \delta)$-far from satisfying $R$.

Observe that in the above definition the value of $\delta$ may be arbitrarily close to 0. If we think of $\delta = 0$ then we get that a graph satisfies $P$ if and only if it satisfies one of the regularity instances of $R$. With this (rough) interpretation in mind, in order to test $P$ one can test the property of satisfying any one of the instances of $R$. Therefore, in some sense we “reduce” the testing of property $P$ to the testing of regularity-instances. We are now ready to state the characterization of the testable graph properties.

**Theorem 8.** A graph property is testable if and only if it is regular-reducible.

The detailed proofs, applications, discussions and further references can be found in [1].

**References**


A (De)constructive Approach to Program Checking (or How to Delegate Work to Your Very Own Adversary)

ShaFi Goldwasser
(joint work with Dan Gutfreund, Alexander Healy, Tali Kaufman), and Guy Rothblum)

One of the main challenges in software engineering is verifying the correctness of software. In the eighties Blum and Kannan [1] proposed the methodology of program “result checking”, which focuses on correctness of the code per input rather than full program verification. The methodology associates every function to be computed with a new piece of code called the checker. Then, given any possibly buggy program for the function and any input, the checker “checks” whether the program on this input computes the function correctly. The work of Blum, Luby, and Rubinfeld [2] further introduced the notion of program testers and correctors. A tester determines whether a given program for a function is correct on random inputs (with relatively high probability). A corrector of a function is given an input and a program that is guaranteed to compute the function correctly on random inputs (but may be buggy on some inputs), and computes (with high probability) the correct output for the given input.

The focus of the rich body of work in the result checking field has been the design of efficient checkers (and tester/correctors) for many specific functions, by exploiting either their algebraic or combinatorial properties. Most notably, these functions include arithmetic operations, matrix operations, and certain graph and group operations. By and large, these are function families which possess random and downwards self-reducibility properties.

This body of work has also found applications beyond the field of program checking. The techniques introduced were pivotal in showing the expressive power of IP and PCP proof systems, and the notion of testers in and of itself has evolved into the successful field of property testing.

Since a correct algorithm for a given function is also trivially a checker for the function, [1] required, in order to avoid triviality, that checkers have the little-oh time property: the running time of the checker must be little-oh of the running time of the most efficient known program that computes the function. An analogue little-oh parallel time property was considered by Rubinfeld [3]: a checker’s parallel running time should be little-oh of the parallel running time of the most efficient known program that computes the function. (Throughout, the standard complexity measure of oracle algorithms is used, where the complexity of the algorithm is measured without the complexity of the oracle’s computations.)

New Work

The work demonstrates new checkers, testers and correctors that are all provably more efficient than the optimal program in terms of circuit depth for the functions at hand. These are designed using a new composition methodology for
improving the circuit depth of checkers, testers and correctors. This approach may, in principal, also be useful to improve other complexity measures.

The idea is the following. We observe that a checker for a function $f$ has access to a potentially powerful resource: the (allegedly correct) program $P$ it is checking, which can often compute a complex function. Our goal is thus to delegate computations from the checker to the program being checked, all the while verifying that the results returned by the delegated computations are correct. To achieve this we start with a checker $C$ for the function at hand – this $C$ may be a previously designed checker, or even just a correct program for the function (which trivially gives a checker) – and then to decompose this checker into sub-computations. The work of these sub-computations is in turn replaced by calls to $P$, the potentially faulty program being checked, on appropriate inputs. This is done by applying a reduction that maps sub-computations to instances of the function $f$ being allegedly computes. The correctness of these delegated sub-computations performed by $P$ is finally verified by checkers for the sub-computations. When the checkers for the sub-computations are more efficient than the sub-computations themselves, this results in a new checker with improved efficiency.

The composition methodology provides a simple way to design checkers that is very similar to the top-down approach of algorithm design: break down the solution of a complex problem into the solution of smaller (and easier) sub-problems, and then combine these solutions, all the while ensuring errors are kept under control. In particular, this approach enables us to construct checkers for functions that do not necessarily have the type of self-reducibility or completeness properties exploited in previous works of [1, 2, 4, 5, 6] as follows.

We first use the Composition Theorem to build checkers that are provably more efficient than the functions they check (in terms of circuit depth) for entire complexity classes, and not just specific functions with special algebraic or combinatorial properties.

**Theorem 1.** For every $i \geq 1$, every language in $\text{RNC}^i$ that is $\text{NC}^1$-hard under $\text{NC}^0$-reductions has a checker in $\text{RNC}^{i-1}$. Every language in $\text{RNC}^i$ that is $\text{NC}^1$-hard under $\text{AC}^0$ reductions has a tester and corrector that are in $\text{RAC}^{i-1}$.

The requirement of being $\text{NC}^1$-hard under $\text{NC}^0$ reductions turns out to not be very restrictive. Examples of natural functions and languages that satisfy the theorem requirements include graph connectivity (in its many variants), deciding whether a given graph has a perfect matching and bounded-degree graph isomorphism, computing the determinant of a matrix, matrix exponentiation, and more.

Next, we turn to the design of parallel depth checkers for matrix functions.

Blum, Luby, and Rubinfeld [2] considered the problem of testing and correcting matrix functions such as multiplication, inverse, determinant and rank. However, they suggested a non-standard model in which the checker/tester/corrector can access (with unit cost) not only the program to be checked, but also a library of (possibly faulty) programs that allegedly compute other related functions. Within this extended model, they show how to test and correct (and thus check) programs for the above matrix functions.
Here, we present standard checkers, testers and correctors for matrix multiplication, inversion, rank and determinant, removing altogether the need for the matrix library model. These checkers/testers/correctors can be implemented in $AC^0$ (and for some ranges of parameters even in $NC^0$). They are provably more efficient than the optimal program for computing these functions in terms of circuit depth. Furthermore, we note that the checkers we build for matrix multiplication and matrix inversion are optimal up to constant factors in every parameter: depth (or parallel time), size (or number of processors) and number of queries.

**Theorem 2.** Matrix multiplication, inversion, determinant and rank have all probabilistic $AC^0$ checkers, testers and correctors. For rank the result holds only over fields that are of size polynomial in the input length. Over a field of cardinality $2^s$ for a constant $s$, matrix multiplication and inversion have probabilistic $NC^0$ checkers, testers and correctors that perform a constant number of calls to the program.

A few additional remarks are in order.

Important building blocks used in applying our composition methodology are efficient checkers for complete languages for low-level complexity classes. Our work shows for the first time how to leverage checkers for complete problems toward the design of checkers for other non-complete problems in the class. Unlike other properties of functions (or languages), the existence of checkers for complete languages did not previously seem to imply or be related to the existence of checkers for non-complete language. Indeed, likely for this reason past work was more concerned with checkers for useful and practical functions and less with checkers for complete languages.

Finally, the paradigm of delegating computation to an untrusted component, which originated in this work, has yielded applications in other settings. In [8] we present new interactive proof systems where very efficient ($NC^0$) verifiers delegate their work to the provers, and new error correcting codes, where much of the decoder’s work is delegated to the encoder (and embedded in the codeword itself). This once again illustrates the fruitful interplay between program checking and other areas in complexity theory.

**References**

Reed-Solomon codes, low-degree automorphisms, and Error-correction with optimal rate

VENKATESAN GURUSWAMI
(joint work with Atri Rudra)

1. Introduction

A fundamental trade-off in coding theory is the one between the percentage of redundancy built into codewords and the fraction of errors that can be corrected. In this talk, we describe an explicit construction of codes that achieves the optimal trade-off between these parameters, along with polynomial time algorithms for performing the error-correction. Formally, for every $0 < p < 1$ and $\varepsilon > 0$, we present an explicit construction of codes with rate $(1 - p - \varepsilon)$ (i.e., which encode $(1 - p - \varepsilon)n$ information symbols into $n$ codeword symbols) that can be list decoded in polynomial time from up to a fraction $p$ of errors. Note that the trade-off we achieve is information-theoretically the best possible, since if the proportion of correctly received symbols is less than the rate, then we have less information at the receiving end than is contained in the message and error-correction is therefore not possible. For low noise levels (small $p$), our codes require almost a factor 2 less redundancy over the previous best result. We stress that our result holds in a worst-case noise model where the channel can corrupt the codeword arbitrarily subject to the bound $p$ on the proportion of errors.

Our codes are simple to describe: they are certain "folded" Reed-Solomon (RS) codes, which are obtained from the classical RS codes via a bundling together of codeword symbols (the resulting encoding is treated as a string over a larger alphabet). A central algebraic idea used in our work is that certain automorphisms of rational function fields induce a low-degree map on the residue of polynomials modulo a certain large degree irreducible. Discovering a similar phenomenon over more general function fields, with the space of sections of some bounded-degree

---

1 Under list decoding, the decoder must output a list of all possible messages that could have been transmitted given that up to a fraction $p$ of errors could have occurred. In order to decode to achieve a rate exceeding $(1 - 2p)$, it is known that the decoder’s worst-case output list size must exceed one. Allowing even moderate sized lists, however, opens up the possibility of achieving a much larger rate, and in particular approaching the information-theoretically optimal bound of $1 - p$. 

---


divisor replacing polynomials, and their evaluations at a large degree “place” replacing residues modulo an irreducible, could lead to exciting new algebraic codes, and perhaps also progress towards achieving list decoding capacity over small, fixed alphabets.

The talk introduced and motivated the problem of list decoding, and then gave a peek into the technical ideas underlying the above result. The algebraic ideas behind these codes, and specifically their precursor, the Parvaresh-Vardy codes [5], have since yielded a new, simple construction of unbalanced bipartite expanders with expansion close to the degree [4]. In turn, this has led to the best known constructions (to date) of randomness extractors. These applications of the Parvaresh-Vardy codes in pseudorandomness were presented in detail in another talk by C. Umans.

We now present some technical details about the folded Reed-Solomon code construction and a formal statement of our main result.

2. Code description and Main result

Consider a Reed-Solomon code $C_{RS}$ consisting of evaluations of degree $k$ polynomials over a finite field $F$ at the set $F^*$ of nonzero elements of $F$. Let $q = |F| = n+1$. Let $\gamma$ be a generator of the multiplicative group $F^*$, and let the evaluation points be ordered as $1, \gamma, \gamma^2, \ldots, \gamma^{n-1}$. Using all nonzero field elements as evaluation points is one of the most commonly used instantiations of Reed-Solomon codes.

Let $m \geq 1$ be an integer parameter called the folding parameter. For convenience, let us assume that $m$ divides $n = q - 1$.

**Definition 1** (Folded Reed-Solomon Code). The $m$-folded version of the RS code $C_{RS}$ is a code of block length $N = n/m$ over $F^m$. The encoding of a message $f(X)$, a polynomial over $F$ of degree at most $k$, has as its $j$th symbol, for $0 \leq j < n/m$, the $m$-tuple $(f(\gamma^{jm}), f(\gamma^{jm+1}), \ldots, f(\gamma^{jm+m-1}))$. In other words, the codewords of the $m$-folded RS code are in one-one correspondence with those of the RS code $C_{RS}$ and are obtained by bundling together consecutive $m$-tuple of symbols in codewords of $C_{RS}$.

The folded version of a RS code thus carries the same information, just “bundled” differently. It is a code of exactly the same rate as the original RS code, but is defined over a larger alphabet. At a high level, folding restricts the flexibility in the subset of evaluation points that an adversary can corrupt.

We now state the main concerning decoding these codes from [2]. In the below statement, the parameter $r$ governs the number of multiplicities at each point imposed during the interpolation (similar to the list decoding algorithm for Reed-Solomon codes from [6, 3]). The parameter $s$ corresponds to the number of dimensions in the interpolation, and the stated bound is obtained through $(s+1)$-variate interpolation.

**Theorem 2.** For every integer $r \geq 1$, every integer $m \geq 1$ and integer $s$, $1 \leq s \leq m$, the folded RS code with the parameters $q,n,N,k$ from Definition 1 can be list
decoded up to a radius
\[ N - \left(1 + \frac{s}{r}\right) \frac{(k^m)^{1/(s+1)}}{m - s + 1} + 2 , \]
in time at most \((nr)^{O(s)}\), and the list size output by the decoder will be at most \(q^s\).

By picking \(r, m\) large enough compared to \(s\), and noting that the rate \(R = k/n\) and \(n = Nm\), the fraction of decoded errors can be made larger than \(1 - (1 + \zeta)R^{1/(s+1)}\) for any desired \(\zeta > 0\). In the limit of large \(s\) (specifically, for \(s = \Theta(\epsilon^{-1} \log(1/R))\)), the decoding radius approaches the list decoding capacity \(1 - R\), leading to the main conclusion of this work:

**Theorem 3** (Explicit capacity-approaching codes). For every \(\epsilon > 0\) and \(0 < R < 1\), there is a family of folded Reed-Solomon codes which have rate at least \(R\) and which can be list decoded up to a fraction \(1 - R - \epsilon\) of errors in time \((N/\epsilon^2)^{O(\epsilon^{-1} \log(1/R))}\) where \(N\) is the block length of the code. The alphabet size of the code as a function of the block length \(N\) is \((N/\epsilon^2)^{O(1/\epsilon^2)}\).

3. Few words on the proof

We now describe at a very high level the key ingredients in proving the above results. For the full details, the reader is referred to [2, 1].

Let us focus on the statement of Theorem 2. By a rather straightforward extension of the bivariate interpolation based decoding paradigm from [6, 3] to the multivariate case, one can compute a non-zero \((s + 1)\)-variate polynomial \(Q(X, Y_1, Y_2, \ldots, Y_s)\) over \(F = F_q\) such that any degree \(k\) polynomial \(f(X)\) whose folded RS encoding is within the radius (1) from the received word must satisfy

\[ Q(X, f(X), f(\gamma X), \ldots, f(\gamma^{s-1} X)) = 0 . \]

The central algebraic step is then to find all degree \(k\) polynomials \(f(X)\) for which (2) holds in an efficient manner, and in the process also prove that there cannot be too many solutions \(f(X)\) to (2). The key insight here is to note the following two facts: (i) The (trivial) identity \(f(\gamma X) \equiv f(X)^{q} \pmod{(X^{q-1} - \gamma)}\) that holds for all polynomials, and (ii) The polynomial \(E(X) = X^{q-1} - \gamma\) is irreducible over \(F_q\). For \(k < q - 1\), all solutions \(f(X)\) of degree at most \(k\) to (2) can thus be found by solving the equation \(Q(X,T,T^q,T^{q^2},\ldots,T^{q^{s-1}}) = 0\) over \(G = F_q[X]/(E(X))\) for \(T\). This amounts to finding the roots of a univariate polynomial over the extension field \(G\), a task that can be performed efficiently.

The algebraic crux above was that the automorphism \(\Gamma\) of the function field \(K = F_q(X)\) induced by \(X \mapsto \gamma X\) satisfied the identity \(\Gamma(f) \mod E(X) = f^{q} \mod E(X)\) for all polynomials \(f\). That is, the automorphism \(\Gamma\) induces a low-degree map w.r.t. the evaluations of polynomials at the “place” corresponding to \(E(X)\).
The use of randomness is widespread in computer science. Many of the best performing algorithms and protocols in many different areas of computer science are randomized. To guarantee their performance these algorithms usually rely on a perfect source of uncorrelated uniformly random bits, yet such a source may not be easy to obtain. We might instead have access to an imperfect random source where the bits are correlated and not uniformly random.

This motivates the study of objects called extractors. Informally, an extractor is an explicit efficiently computable function $\text{Ext} : \{0,1\}^n \rightarrow \{0,1\}^m$ that takes as input bits from an imperfect random source and produces bits that are close to uniformly random (the distance of the output distribution from the uniform distribution is called the error of the extractor). If we had access to such a function, we could use it to extract truly random bits from an imperfect random source. We would then use the extracted bits in our application. Thus we could achieve performance guarantees even with imperfect sources of randomness.

The most general model for a defective source of randomness that has been considered to date is what we will call a weak source [CG88]. The only constraint on a weak source that supplies $n$ total bits is that the probability of getting any particular string from the source is at most $2^{-k}$, where $k$ is called the min-entropy of the source. Such a source is called an $(n,k)$-source. Unfortunately it can be shown that there is no deterministic extractor that can extract from general weak sources.

A natural model for a source that would allow extraction to be feasible is to assume that the source consists of two or more independent parts, each with sufficient entropy. We say that a function $\text{Ext}$ is a $C$-source extractor for entropy $k$ if given any $C$ independent sources with entropy $k$, $X_1, \ldots, X_C$, $\text{Ext}(X_1, \ldots, X_C)$ is close to being uniformly random.
Definition 1. A function \( \text{IndepExt} : (\{0,1\}^n)^C \to \{0,1\}^m \) is an extractor for \( C \) independent sources with min-entropy \( k \) and error \( \epsilon \) if for any independent \((n,k)\) sources \( X^1, \ldots, X^C \) we have that \( \text{IndepExt}(X_1, \ldots, X_C) \) is \( \epsilon \)-close to the uniform distribution in terms of statistical distance.

Another way to view 2-source extractors is as boolean matrices that look random in a strong sense: Every 2-source extractor for entropy \( k \) gives an \( N \times N \) boolean matrix in which every \( K \times K \) minor has roughly the same number of 1’s and 0’s, with \( N = 2^n, K = 2^k \).

The independent sources model is one of the earliest models studied [SV86, Vaz85, CG88]. The probabilistic method shows that most functions are 2-source extractors requiring entropy that is just logarithmic in the total length of each of the sources. Explicit constructions are still very far from achieving this kind of performance. The classical Lindsey Lemma gives a 2-source extractor for sources on \( n \) bits with entropy \( n/2 \). No significant progress was made in improving the entropy requirements over this, until recently. In the last few years, sparked by new results in arithmetic combinatorics [BKT04], there were several results [BIW04, BKS05, Raz05, Bou05].

<table>
<thead>
<tr>
<th>Construction</th>
<th>Min-Entropy ( k )</th>
<th>Output</th>
<th>Error</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>poly(1/(d))-source extractor</td>
<td>( \delta n )</td>
<td>( \Theta(n) )</td>
<td>( 2^{-\Omega(n)} )</td>
<td>[BIW04]</td>
</tr>
<tr>
<td>3-source extractor</td>
<td>( \delta n ), any constant ( \delta )</td>
<td>( \Theta(1) )</td>
<td>( O(1) )</td>
<td>[BKS+05]</td>
</tr>
<tr>
<td>3-source extractor</td>
<td>One source: ( \delta n ), any constant ( \delta ). Other sources may have ( \text{polylog}(n) ) min-entropy.</td>
<td>( \Theta(1) )</td>
<td>( O(1) )</td>
<td>[Raz05]</td>
</tr>
<tr>
<td>2-source extractor</td>
<td>One source: ( (0.5 + \alpha)n ), ( \alpha &gt; 0 ). Other source may have ( k = \text{polylog}(n) ) min-entropy.</td>
<td>( \Theta(k) )</td>
<td>( 2^{-\Omega(k)} )</td>
<td>[Raz05]</td>
</tr>
<tr>
<td>2-source extractor</td>
<td>( (0.5 - \alpha_0)n ) for some universal constant ( \alpha_0 &gt; 0 )</td>
<td>( \Theta(n) )</td>
<td>( 2^{-\Omega(n)} )</td>
<td>[Bou05]</td>
</tr>
<tr>
<td>( O(1/\delta) )-source extractor</td>
<td>( n^{\delta} )</td>
<td>( \Theta(k) )</td>
<td>( k^{-\Omega(1)} )</td>
<td>[Rao06]</td>
</tr>
<tr>
<td>( O(1/\delta) )-source extractor</td>
<td>( n^{\delta} )</td>
<td>( \Theta(k) )</td>
<td>( 2^{-k^{\Omega(1)}} )</td>
<td>[BRSW06]</td>
</tr>
<tr>
<td>3-source extractor (with constraints on input lengths)</td>
<td>( n^{\delta} ) for any constant ( \delta ) (additional constraints apply)</td>
<td>( k - o(k) )</td>
<td>( 2^{-k^{\Omega(1)}} )</td>
<td>[LRZ07]</td>
</tr>
</tbody>
</table>

Table 1. Performance of recent extractors for independent sources.
Results in the Talk. We give a polynomial time computable extractor that extracts $k$ random bits from $O(k^{\log n})$ independent $(n,k)$-sources with error $k^{-c}$ for any $k(n) > \log^\gamma n$ and some universal constant $c > 1$. An interesting setting of parameters is when $k = n^\gamma$ for some $0 < \gamma < 1$. In this case we get an extractor for a constant number of sources that extracts a constant fraction of the total min-entropy with exponentially small error.

Techniques. Many extractor constructions in the past have been based on the paradigm of iterative condensing [RSW00, TUZ01, CRVW02, LRVW03, BIW04]. The idea is to start with some distribution that has low min-entropy and apply a function (called a condenser) whose output has a better min-entropy rate. Repeating this process, we eventually obtain a distribution which has very high min-entropy rate. Then we can apply some other extractor which works for such a high min-entropy rate to obtain random bits. The extractor in this paper can also be viewed as an example of this paradigm, with a slight twist.

We make progress by considering a more restricted model for sources called somewhere random [TS96]. A source is somewhere random if it samples from some distribution on boolean matrices, such that at least one of the rows of the matrix is distributed uniformly. An important concept that we introduced in that chapter is that of aligned somewhere random sources. Two somewhere random sources with the same number of rows are said to be aligned if there is an $i$ such that the $i$th row of both sources are distributed uniformly.

We think of the number of rows of a somewhere random sources as a measure of the quality of the source. The fewer the number of rows, the better the quality is. Our construction works by iteratively improving the quality (reducing the number of rows) of the somewhere random sources that we are working with until extracting randomness from them becomes easy.

References


Unbalanced Expanders and Randomness Extractors from Parvaresh-Vardy Codes

CHRIS UMAN

(joint work with Venkatesan Guruswami, Salil Vadhan)

A long line of work has been devoted to obtaining explicit constructions of randomness extractors, defined below:

**Definition 1.** A function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k,\epsilon)$ extractor if for every $X$ with minentropy at least $k$, $E(X,Y)$ is $\epsilon$-close to uniform, when $Y$ is uniformly distributed on $\{0,1\}^d$. An extractor is explicit if it is computable in polynomial time.

Many applications have been sought for these objects in a diverse range of settings, and consequently a lot of effort has been spent trying to find explicit constructions. An extractor construction matching the non-constructive bounds would have a seed length of $d = \log n + 2\log(1/\epsilon) + O(1)$ and an output length of $m = k + d - 2\log(1/\epsilon) - O(1)$. We still do not have explicit constructions of optimal extractors; the best construction prior to this work was by Lu, Reingold, Vadhan, and Wigderson [LRVW03], who get within constant multiplicative factors.
of optimal in both the seed length and the output length (for $\epsilon$ that is not too small).

In this talk we describe a new explicit construction of extractors that matches [LRVW03], and also handles small $\epsilon$. One of the main advantages of our result is the simplicity of the construction and its proof.

The construction works by first building an intermediate object called a lossless condenser, defined below:

**Definition 2.** A function $C : \{0, 1\}^n \times \{0, 1\}^d \to \{0, 1\}^m$ is an $k \to k'$ condenser if for every $X$ with minentropy at least $k$, $C(X, Y)$ is $\epsilon$-close to a distribution with minentropy $k'$, when $Y$ is uniformly distributed on $\{0, 1\}^d$. A condenser is explicit if it is computable in polynomial time. A condenser is called lossless if $k' = k + d$.

Nonconstructively, there exist lossless condensers with seed length $d = \log n + \log(1/\epsilon) + O(1)$ and output length $m = k + d + \log(1/\epsilon) + O(1)$. The best previous constructions of lossless condensers were by Ta-Shma, Umans, and Zuckerman [TUZ01], with a further improvement in [TU06]. These construction get within a constant multiplicative factor of optimal in either of the two parameters (seed length and output length), at the expense of being a super-constant multiplicative factor away from optimal in the other. Our main construction achieves a lossless condenser that is within a constant factor of optimal in both parameters simultaneously.

It is most natural to describe our main construction as an explicit construction of yet another object, an (unbalanced, bipartite) expander graph, defined below:

**Definition 3.** A bipartite (multi)graph $G = (U, V, E)$ is a $(K, A)$ expander if for every set $S \subseteq U$ of size $K$, we have $|E[S]| \geq A \cdot K$.

It turns out that a $(K = 2^k, (1 - \epsilon)2^d)$ expander is equivalent to a $k \to k + d$ (lossless) condenser with seed length $d$ [TUZ01]. In the language of expanders, our main construction achieves expansion $(1 - \epsilon)D$ while the degree $D$ is a polynomial in the optimal $O(\log N/\epsilon)$ (for the unbalanced case), and the right-hand side has size that is also a polynomial in the optimal $O(KD/\epsilon)$.

Our main construction is based on Parvaresh-Vardy codes [PV05], and its proof essentially amounts to a tight analysis of the so-called “list-recovering” properties of these codes, which follows the list-decoding analysis of [PV05] closely. Let $F_q$ be the field with $q$ elements, and let $h < q$ be a parameter we will set later. Let $E(Y)$ be a degree $n$ polynomial that is irreducible over $F_q$. We identify the left-hand-side of the bipartite graph with the set of degree $n - 1$ univariate polynomials over $F_q$. Given such a polynomial $f(Y)$, define $f_i(Y)$ to be $f(Y)^i \mod E(X)$. For each field element $y \in F_q$, the $y$-th neighbor of $f$ is $(y, f_0(y), f_1(y), \ldots, f_m(y)) \in F_q^{m+1}$. This is just $y$ prepended to the $y$-th coordinate of the Parvaresh-Vardy codeword corresponding to $f$.

The natural way to prove expansion would be to fix a subset of size $K$ of the left-hand-side, and argue that its neighbor set has size at least $AK$. An equivalent
proof strategy, that will be useful here, fixes a subset $T$ of size $AK - 1$ of the right-hand-side, and argues that that the set $LIST(T)$, consisting of all left-hand-side vertices whose neighbor set is entirely contained within $T$, has size at most $K - 1$.

Now we can present the proof, which is algebraic, short, and self-contained.

The main lemma is:

**Lemma 4.** For $A = q - nmh$, and $K = h^m$, if $T \subseteq \mathbb{F}_q^{n+1}$ has size $AK - 1$ then $LIST(T) \subseteq \mathbb{F}_q^n$ has size at most $K - 1$.

**Proof.** Fix a set $T \subseteq \mathbb{F}_q^n$. Let $Q(Y, Z_0, \ldots, Z_{m-1})$ be a non-zero polynomial that vanishes on $T$, with $\deg(Y) \leq A - 1$, and $\deg(Z_i) \leq h - 1$. We assume that $E(Y)$ does not divide $Q$, without loss of generality.

Now consider a degree $n - 1$ polynomial $f$ in $LIST(T)$. By definition all of its neighbors are in $T$, and so $Q(y, f_0(y), \ldots, f_{m-1}(y)) = 0$. But this univariate polynomial in $y$ has degree less than $q$, and it vanishes on $q$ points, so it must be the zero polynomial. Substituting the definition of the $f_i$, we have

$$Q(Y, (f(Y) \mod E(Y)), (f(Y)^h \mod E(Y)), \ldots, (f(Y)^{h^{m-1}} \mod E(Y))) \equiv 0.$$ 

and the same holds after taking the left-hand-side modulo $E(Y)$.

Thus, over the extension field $\mathbb{F}_q[Y]/E(Y)$ (in which $f(Y)$ is a field element), we have $Q(Y, f(Y), f(Y)^h, \ldots, f(Y)^{h^{m-1}}) = 0$, or equivalently, that $f$ is a root of the univariate polynomial $Q'(Z) = Q(Y, Z, Z^h, \ldots, Z^{h^{m-1}}) \mod E(Y)$. Finally, the degree of $Q'$ is at most $(h - 1)(1 + h + h^2 + \ldots + h^{m-1}) = h^m - 1$, which is an upper bound on the size of $LIST(T)$. \hfill \square

A minor tweak to this proof gives an upper bound of $K' - 1$ on $|LIST(T)|$, given a $T$ of size $AK' - 1$, for any $K' \leq K$. Setting $h = (nm/e)^{1/\alpha}$ and $q = h^{1+\alpha}$ delivers the promised expansion, degree, and right-hand-side size.

Viewed as a condenser, this construction condenses an arbitrary source losslessly into one with entropy rate any constant arbitrarily close to 1. We can apply known extractors [Zuc97] to such a source to extract a constant fraction of the minentropy; for constant $\epsilon$, we can even use the well-known “expander-walk” extractor which is very simple to describe and analyze with Chernoff-type bounds for expander walks [Gi98].

Other ideas from list-decodable error-correcting codes are useful in our setting as well. The “repeated roots” idea from [GS99] can be used to reduce the seed length (at the expense of some entropy loss). Ideas from [GR06] can be used to show that a very natural construction based on Reed-Solomon codes yields a good condenser. Specifically, the $y$-th neighbor is now $(f(y), f(\alpha y), \ldots, f(\alpha^{m-1} y)) \in \mathbb{F}_q^n$, where $\alpha$ is a generator of $\mathbb{F}_q^*$, and essentially the same proof technique shows that this is a condenser whose output retains a $(1 - \delta)$ fraction of the entropy, for any $\delta > 0$.

In summary, we give the best explicit constructions of randomness extractors and unbalanced bipartite expander graphs (a.k.a. “lossless condensers”) to date, using a simple construction and proof technique based on Parvaresh-Vardy codes.
Complexity Theory

REFERENCES


Statistically Hiding Commitments from Any One-Way Function

SALIL VADHAN

(joint work with I. Haitner, M. Nguyen, S.J. Ong, O. Reingold)

As first discovered by Shannon [22] for the case of encryption, most interesting cryptographic tasks are impossible to achieve with absolute, information-theoretic security. Thus, modern cryptography aims to design protocols that are computationally intractable to break. Specifically, following Diffie and Hellman [5], this is typically done by showing that breaking the protocol is as hard as some intractable problem from complexity theory. Unfortunately, proving lower bounds of the sort needed seems beyond the reach of current techniques in complexity theory, and indeed would require at least proving $P \neq NP$.

Given this state of affairs, research in the foundations of cryptography has aimed to design cryptographic protocols based on complexity assumptions that are as weak and general as possible. This project was enormously successful in the 1980’s. In a beautiful sequence of works, it was shown that many cryptographic primitives, such as pseudorandom generators, pseudorandom functions, private-key encryption and authentication, digital signatures, (computationally hiding) bit commitment, and (computational) zero-knowledge proofs could be constructed from any one-way function [12, 6, 21, 16, 8], and moreover this complexity assumption is minimal in the sense that each of these primitives (and indeed almost any cryptographic task) implies the existence of one-way functions [13, 20]. Moreover,
it was shown that many of the remaining primitives, such as public-key encryption, collision-resistant hashing, and oblivious transfer, could not be reduced to the existence of one-way functions in a “black-box” manner [14, 23].

However, a few important primitives have resisted classification into the above categories. That is, it is only known how to build these primitives from seemingly stronger assumptions than the existence of one-way functions, yet there is no black-box separation between these primitives and one-way functions. In this work, we are interested in statistically hiding and computationally binding commitment schemes.

Statistically Hiding Commitments. A commitment scheme defines a two-stage interactive protocol between a sender $S$ and a receiver $R$; informally, after the commit stage, $S$ is bound to (at most) one value, which stays hidden from $R$, and in the reveal stage $R$ learns this value. The two security properties hinted at in this informal description are known as binding (namely, that $S$ is bound to at most one value after the commit stage) and hiding (namely, that $R$ does not learn the value to which $S$ commits before the reveal stage). In a statistically hiding computationally-binding commitment scheme the hiding property holds even against all-powerful receivers (i.e., hiding holds information-theoretically), while the binding property is required to hold only for polynomially-bounded senders.

Statistical commitment schemes can be used as a building block in constructions of statistical zero-knowledge arguments [3, 17] and certain coin-tossing protocols [15]. It therefore implies, via standard reduction, a way to transform a large class of protocols that are secure assuming an all-powerful honest-but-curious party, into one that is secure even when this party maliciously deviates from the protocol. More generally, when used within protocols in which certain commitments are never revealed, statistical commitments have the following advantage over computationally-binding commitment schemes: in such a scenario, it needs only be infeasible to violate the binding property during the period of time the protocol is run, whereas the committed values will remain hidden forever (i.e., regardless of how much time the receiver invests after completion of the protocol).

Perfectly-hiding1 commitment schemes were first shown to exist based on specific number-theoretic assumptions [1, 3] or, more generally, based on any collection of claw-free permutations [9] with an efficiently-recognizable index set [7]. Statistical commitment schemes can also be constructed from collision-resistant hash functions [4, 18]. Naor et al. [17] showed a construction of a perfectly-hiding commitment scheme based on any one-way permutation. Haitner et. al. [10] make progress by constructing statistical commitment based on regular one-way functions and also on the so called approximable-size one-way functions. The question of whether one-way functions imply statistical commitments, however, was left open.

1Very informally, in a statistically-hiding commitment scheme the receiver learns only a negligible amount of information about the sender’s committed value, whereas in a perfectly-hiding commitment scheme the receiver learns nothing. Note that any perfectly-hiding scheme is trivially also statistically hiding.
We mention that the complementary notion of commitment schemes, where the hiding is computational and the binding holds even w.r.t. an all powerful sender, was already known to be implied by the existence of one-way functions [12, 16].

**Our result.** Our main result is that the existence of one-way functions is a sufficient condition for the existence of statistical commitment. Namely, we prove the following theorem.

**Theorem 1.** Assuming that one-way functions exist, then there exists a statistically-hiding and computationally-binding commitment scheme.

By Impagliazzo and Luby [13], the existence of statistical commitment schemes implies the existence of one-way functions and thus the above result is tight.

One of the main applications of statistically hiding commitments are statistical zero-knowledge arguments [2, 8, 3]. These are zero-knowledge protocols for proving membership in any NP language where the zero knowledge property is statistical (i.e. even a computationally unbounded verifier learns nothing from the protocol) and the soundness is computational (i.e. no polynomial-time prover can convince the verifier of a false assertion, except with negligible probability). Thus, we also deduce:

**Theorem 2.** Assuming that one-way functions exist, then every language in NP has a statistical zero-knowledge argument system.

**References**


Correlation bounds are fundamental in computational complexity. Proving that \( \text{Cor}(f, C) < 1 \) is equivalent to establishing that \( f \notin C \), but what is far more desired is proving that \( \text{Cor}(f, C) \) is very close to zero, for natural functions \( f \) and complexity classes \( C \). Such bounds yield pseudorandom generators that "fool" the class \( C \) (e.g., \([10, 11, 9, 16])\), and they also imply lower bounds for richer classes related to \( C \) (e.g., if \( \text{Cor}(f, C) < 1/t \) then \( f \) cannot be computed exactly by any function which is the majority of any \( t \) functions from \( C \) [4]). For these applications, we would like to prove correlation bounds as close to zero as possible.

A celebrated way of decreasing correlation (a.k.a. amplifying hardness) is via an XOR lemma, first suggested by Yao in his seminal paper [18] (cf. [3]). One starts with a function \( f \) of nontrivial correlation with \( C \), and constructs a new function \( f^{\oplus m} \) (on \( n \cdot m \) bits), which is the exclusive-OR of the value of \( f \) on \( m \) independent inputs. The hope is that the correlation will decay exponentially with \( m \). This

\begin{align*}
\text{Cor}(f, p) &:= |E_x[f(x) \cdot p(x)]| = \Pr_x[f(x) = p(x)] - \Pr_x[f(x) \neq p(x)] \\
&\in [0, 1].
\end{align*}

For a complexity class \( C \) (e.g., circuits of size \( s \) on \( n \) bits), we denote by \( \text{Cor}(f, C) \) the maximum of \( \text{Cor}(f, p) \) over all functions \( p \in C \). In other words, \( \text{Cor}(f, C) \) captures how well on average can we compute \( f \) using a function from \( C \).

A natural measure of agreement between two functions is their correlation, which measures the agreement on a random input. Formally, the correlation between two functions \( f, p \in \{0, 1\}^n \rightarrow \{-1, 1\} \) is defined as

\[ \text{Cor}(f, p) := |E_x[f(x) \cdot p(x)]| = \Pr_x[f(x) = p(x)] - \Pr_x[f(x) \neq p(x)] \in [0, 1]. \]

For a complexity class \( C \) (e.g., circuits of size \( s \) on \( n \) bits), we denote by \( \text{Cor}(f, C) \) the maximum of \( \text{Cor}(f, p) \) over all functions \( p \in C \). In other words, \( \text{Cor}(f, C) \) captures how well on average can we compute \( f \) using a function from \( C \).

Norms, XOR lemmas, and lower bounds for \( GF(2) \) polynomials and multiparty protocols

Avi Wigderson

(joint work with Emanuele Viola)
idea is best demonstrated in the information-theoretic setting, in which we try to compute the value of a biased coin. In our language, take $C$ to be the class of constant functions, and $f$ any function with $|E_x[f(x)]| = \text{Cor}(f, C) = \epsilon$. Then it is easy to see that $\text{Cor}(f^m, C) = \epsilon^m$ for every $m$.\footnote{Strictly speaking, now $C$ denotes the constant functions on $n \cdot m$ bits.} So the decay of the correlation in this trivial scenario is purely exponential in the number of copies $m$.

Yao's XOR lemma deals with the most studied computational model, namely polynomial-size circuits, and goes as follows. Let $C$ be the class of Boolean circuits of size $s$, and let $f$ be any function on $n$ bits with $\text{Cor}(f, C) \leq \epsilon$. Then for any large $m$ and small $\alpha > 0$, if $C'$ is the class of circuits of size $s \cdot (\alpha/nm)^2$ then $\text{Cor}(f^m, C') \leq \epsilon^m + \alpha$. Many proofs of this XOR lemma were given, starting with Levin [8, 5, 3, 6]. All in fact show that if this lemma holds in more general circumstances, namely as long as $C$ can compute the majority of functions in $C'$.

However, none of these proofs can be applied to the computational models for which we actually can establish the existence of functions with non-trivial correlation (i.e., prove lower bounds), such as low-degree $GF(2)$ polynomials, multiparty protocols, or constant-depth circuits (cf. [14, Chapter 6]). Specifically, none of the above proofs can be applied to obtain a correlation bound of $1/n$ for a function on $n$ bits. Another weakness of the results in [8, 5, 3, 6] is their loss in resources (e.g., circuit size) in $C'$ with respect to $C$ (cf. [3]).

**Our results.** In this paper we prove new XOR lemmas for two models: low-degree polynomials over $GF(2)$, and low-communication multiparty protocols.

We show that if a Boolean function has correlation at most $\epsilon \leq 1/2$ with any of these models, then the correlation of the parity of its values on $m$ independent instances drops exponentially with $m$. More specifically:

- For $GF(2)$ polynomials of degree $d$, the correlation drops to $\exp(-m/4^d)$. No XOR lemma was known even for $d = 2$.
- For $c$-bit $k$-party protocols, the correlation drops to $2^c \cdot \epsilon^{m/2^k}$. No XOR lemma was known for $k \geq 3$ parties.

Another contribution in this paper is a general derivation of direct product lemmas from XOR lemmas. In particular, assuming that $f$ has correlation at most $\epsilon \leq 1/2$ with any of the above models, we obtain the following bounds on the probability of computing $m$ independent instances of $f$ correctly:

- For $GF(2)$ polynomials of degree $d$ we again obtain a bound of $\exp(-m/4^d)$.
- For $c$-bit $k$-party protocols we obtain a bound of $2^{-\Omega(m)}$ in the special case when $\epsilon \leq \exp(-c \cdot 2^k)$. In this range of $\epsilon$, our bound improves on a direct product lemma for two parties by Parnafes, Raz, and Wigderson [12].

We also give improved (or just simplified) lower bounds in these models. In particular we give a new proof that the $\text{Mod}_m$ function on $n$ bits, for odd $m$, has correlation at most $\exp(-n/4^d)$ with degree-$d$ $GF(2)$ polynomials.

Both proofs of our XOR lemmas use a common approach, very different from the one used for circuits. To each of these complexity classes $C$ we associate a real
norm $N$ on all Boolean functions which has the following properties (informally stated):

1. **$N$ captures correlation with $C$.** For every function $f$, $N(f) \approx \text{Cor}(f, C)$.
2. **$N$ commutes with XOR.** Let $f, g$ be two functions on disjoint inputs, then $N(f \cdot g) = N(f) \cdot N(g)$.

Given such a norm $N$, the proof of an XOR lemma for $C$ is straightforward:

$$\text{Cor}(f^\otimes m, C) \approx N(f^\otimes m) = N(f)^m \approx \text{Cor}(f, C)^m.$$

Of course, the challenge is to find the appropriate norm and prove their properties. We explain how such norms are related to the basic question of “Property Testing,” where the problem being tested is how close is the given function $f$ to the class $C$.

We also explain how such norms suggest themselves when the class $C$ in question is a linear space. This is the case with $GF(2)$ polynomials, in which case indeed the well-known Gowers’ norms are used. For multi-party protocols, which are not a linear space, we explain how they can be “approximated” by a linear space, for which a natural norm is implicit in previous papers.

**References**


Towards 3-Query Locally Decodable Codes of Subexponential Length

SERGEY YEKHANIN

1. Abstract

A q-query Locally Decodable Code (LDC) encodes an n-bit message x as an N-bit codeword C(x), such that one can probabilistically recover any bit x_i of the message by querying only q bits of the codeword C(x), even after some constant fraction of codeword bits has been corrupted.

We give new constructions of three query LDCs of vastly shorter length than that of previous constructions. Specifically, given any Mersenne prime \( p = 2^t - 1 \), we design three query LDCs of length \( N = \exp(n^{1/5}) \), for every n. Based on the largest known Mersenne prime, this translates to a length of less than \( \exp(n^{10^{-1}}) \), compared to \( \exp(n^{1/2}) \) in the previous constructions.

2. Introduction

Classical error-correcting codes allow one to encode an n-bit string x into a N-bit codeword C(x), in such a way that x can still be recovered even if C(x) gets corrupted in a number of coordinates. For instance, codewords C(x) of length \( N = O(n) \) already suffice to correct errors in up to \( \delta N \) locations of C(x) for any constant \( \delta < 1/4 \). The disadvantage of classical error-correction is that one needs to consider all or most of the (corrupted) codeword to recover anything about x. Now suppose that one is only interested in recovering one or a few bits of x. In such case more efficient schemes are possible. Such schemes are known as locally decodable codes (LDCs). Locally decodable codes allow reconstruction of an arbitrary bit x_i, from looking only at q randomly chosen coordinates of C(x),
where $q$ can be as small as 2. Locally decodable codes have found numerous applications in complexity theory and cryptography. See [10], [5] for a survey. Below is a slightly informal definition of LDCs:

A $(q, \delta, \epsilon)$-locally decodable code encodes $n$-bit strings to $N$-bit codewords $C(x)$, such that for every $i \in [n]$, the bit $x_i$ can be recovered with probability $1 - \epsilon$, by a randomized decoding procedure that makes only $q$ queries, even if the codeword $C(x)$ is corrupted in up to $\delta N$ locations.

One should think of $\delta > 0$ and $\epsilon < 1/2$ as constants. The main parameters of interest in LDCs are the length $N$ and the query complexity $q$. Ideally we would like to have both of them as small as possible. The notion of locally decodable codes was explicitly discussed in various places in the early 1990s. Katz and Trevisan [6] were the first to provide a formal definition of LDCs and prove lower bounds on their length. Further work on locally decodable codes includes [2, 7, 11]. The length of optimal 2-query LDCs was settled by Kerenidis and de Wolf in [7] and is $\exp(n)$. The length of optimal 3-query LDCs is unknown. The best upper bound prior to our work was $\exp(n^{1/2})$ due to Beimel et al., and the best lower bound is $\tilde{\Omega}(n^2)$ [7, 14]. For general (constant) $q$ the best upper bound was $\exp(n^{O(\log \log n/(q \log q)))}$ due to Beimel et al. [2] and the best lower bound is $\Omega(n^{1/1/(q/2)+1})$ [7, 14].

3. Our results

We give new families of locally decodable codes whose length is vastly shorter than that of previous constructions. We show that every Mersenne prime $p$ (i.e., a prime of the form $p = 2^t - 1$) yields a family of three query locally decodable codes of length $\exp(n^{1/d})$. The largest Mersenne prime known currently has $t = 32,582,657 > 10^9$. Substituting this prime into our theorem we conclude that for every $n$ there exists a three query locally decodable code of length $\exp(n^{1/32,582,657})$.

It has often been conjectured that the number of Mersenne primes is infinite. If indeed this conjecture holds, our constructions yield three query locally decodable codes of length $N = \exp(n^{O(\log \log n)})$ for infinitely many $n$. Finally, assuming that the conjecture of Lenstra, Pomerance and Wagstaff [12, 9, 13] regarding the density of Mersenne primes holds, our constructions yield three query locally decodable codes of length $N = \exp(n^{O(\log^{-1/\epsilon} \log n)})$ for all $n$, for every $\epsilon > 0$.

4. Our technique

All previously known constructions of locally decodable codes and private information retrieval schemes are (implicitly or explicitly) centered around the idea of representing a message $x$ by an evaluation of a certain low degree polynomial over a finite field. Our constructions take a completely different approach. We start by reducing the problem of constructing locally decodable codes to the problem of
designing certain families of sets with restricted intersections. We use elementary algebra over finite fields to design such families.

The heart of our construction is the design of a set $S \subseteq \mathbb{F}_p^n$ for a prime $p$ that simultaneously satisfies two properties: (1) There exist two large sequences of vectors $u_1, \ldots, u_n, v_1, \ldots, v_n$ in some low dimensional space $\mathbb{F}_p^m$, such that the dot products $\langle u_i, v_i \rangle = 0$ for all $i$, and the dot products $\langle u_i, u_j \rangle \in S$ for all $i \neq j$. We refer to this property as the combinatorial niceness of $S$; (2) For a small integer $q$ there exists a $q$ sparse polynomial $\phi(x) \in \mathbb{F}_q[x]$ such that the common GCD of all polynomials of the form $\phi(x^\beta)$, $\beta \in S$ and the polynomial $x^p - 1$ is non-trivial. We refer to this property as the algebraic niceness of $S$. Our notion of combinatorial niceness is related to the notion of set families with restricted intersections in [1].

Our construction of locally decodable codes thus comes in three steps: First we show that a set $S$ exhibiting both combinatorial and algebraic niceness leads to good locally decodable codes. In particular the length $n$ of the sequences $u_1, \ldots, u_n$ and $v_1, \ldots, v_n$ corresponds to the number of message bits we can encode, while the length of the codewords we build is $N = p^m$. So the longer the sequence and the smaller the dimension the better. The query complexity of our codes is given by the parameter $q$ from the definition of algebraic niceness of $S$. This step of our construction is quite general and applies to vectors $u_1, \ldots, v_n$ and subsets $S$ over any field. It leads us to the task of identifying good sets that are both combinatorially and algebraically nice, and these tasks narrow our choice of fields. As our second step we focus on combinatorial niceness. In general big sets tend to be “nicer” (allow longer sequences) than small ones. We show that every multiplicative subgroup of a prime field is combinatorial as nice as its cardinality would allow. This still leaves us with a variety of fields and subsets to work with. Finally as the last step we attempt to understand the algebraic niceness of sets. We focus on the very narrow case of Mersenne primes $p$ and the subgroup generated by the element 2 in $\mathbb{F}_p^*$. We manage to show that this subgroup is nice enough to get 3-query locally decodable codes, leading to our final result. A formal treatment of our constructions can be found in [15].

References

We study the following natural question: Which cryptographic primitives (if any) can be realized by functions with constant input locality, namely functions in which every bit of the input influences only a constant number of bits of the output? This continues the study of cryptography in low complexity classes. It was recently shown [1] that, under standard cryptographic assumptions, most cryptographic primitives can be realized by functions with constant output locality, namely ones in which every bit of the output is influenced by a constant number of bits from the input.

We (almost) characterize what cryptographic tasks can be performed with constant input locality. On the negative side, we show that primitives which require some form of non-malleability (such as digital signatures, message authentication, or non-malleable encryption) cannot be realized with constant input locality. On the positive side, assuming the intractability of certain problems from the domain of error correcting codes (namely, hardness of decoding a random linear code or the security of the McEliece cryptosystem), we obtain new constructions of one-way functions, pseudorandom generators, commitments, and semantically-secure public-key encryption schemes whose input locality is constant. Moreover, these constructions also enjoy constant output locality. Therefore, they give rise to cryptographic hardware that has constant-depth, constant fan-in and constant fan-out. As a byproduct, we obtain a pseudorandom generator whose output and input locality are both optimal (namely, 3).
Lower Bounds on Signatures From Symmetric Primitives

Boaz Barak
(joint work with Mohammad Mahmoody-Ghidary)

We show that every black-box construction of one-time signature schemes from a random oracle achieves security at most $\text{poly}(q)^{2^q}$, where $q$ is the total number of queries to the oracle by the generation, signing, and verification algorithms. That is, any such scheme can be broken with probability close to $1$ by a (computationally unbounded) adversary making $\text{poly}(q)^{2^q}$ queries to the oracle. This is tight up to a constant factor in the number of queries, since a simple modification of Lamport’s scheme [2] achieves $2^{(2^{1/n})}$ security using $q$ queries. Our result provides the first lower bound on the efficiency of constructing signature schemes using a random oracle. A previous result by Gennaro et al [1] gave a lower bound on such constructions using (highly non-random) one-way functions.

Our results extend (with a loss of a constant factor in the number of queries) also to the random permutation and ideal-cipher oracles, and so can be taken as evidence of an inherent efficiency gap between signature schemes and symmetric primitives such as block ciphers, hash functions, and message authentication codes.

References


Lower Bounds for Randomized Read/Write Stream Algorithms

Paul Beame
(joint work with T. S. Jayram, Atri Rudra)

Motivated by the capabilities of modern storage architectures, we consider the following generalization of the data stream model where the algorithm has sequential access to multiple streams. Unlike the data stream model, where the stream is read only, in this new model (introduced in [1, 2]) the algorithms can also write onto streams. There is no limit on the size of the streams but the number of passes made on the streams is restricted. On the other hand, the amount of internal memory used by the algorithm is scarce, similar to data stream model.

We resolve the main open problem in [3] of proving lower bounds in this model for algorithms that are allowed to have 2-sided error. Previously, such lower
bounds were shown only for deterministic and 1-sided error randomized algorithms [2, 3]. We consider the classical set disjointness problem that has proved to be invaluable for deriving lower bounds for many other problems involving data streams and other randomized models of computation. For this problem, we show a near-linear lower bound on the size of the internal memory used by a randomized algorithm with 2-sided error that is allowed to have $o(\log N/\log \log N)$ passes over the streams. This bound is almost optimal since there is a simple algorithm that can solve this problem using logarithmic memory if the number of passes over the streams is allowed to be $O(\log N)$.

Applications include near-linear lower bounds on the internal memory for well-known problems in the literature: (1) approximately counting the number of distinct elements in the input ($F_0$); (2) approximating the frequency of the mode of an input sequence ($F_{\infty}$); (3) computing the join of two relations; and (4) deciding if some node of an XML document matches an XQuery (or XPath) query.

Our techniques involve a novel direct-sum type of argument that yields lower bounds for many other problems. Our results asymptotically improve all previously known bounds for problems in the read/write streams model even in deterministic and 1-sided error models of computation.

References


On the complexity of graph polynomials

MARKUS BLÄSER

(joint work with Holger Dell, Mahmoud Fouz, 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 $\xi$ and map $G$ to $P_G(\xi)$, we get a new graph invariant that maps graphs to elements of $R$. We deal with the following question: What is the complexity of this mapping in dependence on $\xi$? A famous result by Jaeger, Vertigan and Welsh [2] says that for almost all points (in the Zariski sense), it is $\#P$-hard to evaluate the Tutte polynomial at this point. We show a similar result for cover polynomial, an equivalent of the Tutte polynomial for directed graphs (see also [1]), and for the interlace polynomial. Since all these polynomials are definable in monadic second
order logic, this gives rise to the following “difficult point conjecture”, cf. [3]:
For every graph polynomial that is definable in monadic second order logic and
that is \#P-hard to evaluate for at least one point, its evaluation is \#P-hard for
almost all points. Finally, we propose that one should study graph polynomials
in an algebraic model a la Blum, Shub, and Smale, since most graph polynomials
are defined over \( \mathbb{C} \). Prior studies usually looked only at fields that have discrete
representations like algebraic field extensions of \( \mathbb{Q} \).

References
[1] Markus Bläser, Holger Dell, Complexity of the cover polynomial, Proc. 34th Int. Coll. on
Automata, Languages, and Programming (ICALP), Springer Lecture Notes in Computer
Science 4596 (2007).
[2] F. Jaeger, D.L. Vertigan, D.J. Welsh, On the computational complexity of the Jones and
Tutte polynomials, Mathematical Proceedings of the Cambridge Philosophical Society , 108
(1990), 35–52.
[3] Johann A. Makowsky, From a zoo to a zoology: towards a general theory of graph polyno-

Sampling methods for shortest vectors, closest vectors and successive
minima
JOHANNES BLÖMER
(joint work with Stefanie Naewe)

We study four problems from the geometry of numbers, the shortest vector problem
(SVP), the closest vector problem (CVP), the successive minima problem (SIVP),
and the shortest independent vectors problem (SIVP). Extending and generalizing
results of Ajtai, Kumar, and Sivakumar we present probabilistic single exponential
time algorithms for all four problems for all \( \ell_p \) norms. The results on SMP and
SIVP are new for all norms. The results on SVP and CVP generalize previous
results of Ajtai et al. for the Euclidean \( \ell_2 \) norm to arbitrary \( \ell_p \) norms. We
achieve our results by introducing a new lattice problem, the subspace avoiding
problem (SAP). We describe a single exponential time algorithm for SAP. We
also describe polynomial time reductions from SVP, CVP, SMP, and SIVP to SAP,
establishing the single exponential time algorithm for the four classical lattice
problems. This approach leads to a unified algorithmic treatment of the lattice
problems SVP, CVP, SMP, and SIVP.

Pseudorandom generators for low degree polynomials
ANDREJ BOGDANOV
(joint work with Emanuele Viola)

We present a new approach to constructing pseudorandom generators that fool
low-degree polynomials over finite fields, based on the Gowers norm. Using this
approach, we obtain the following main constructions of explicitly computable generators $G : \mathbb{F}^s \to \mathbb{F}^n$ that fool polynomials over a prime field $\mathbb{F}$:

1. a generator that fools degree-2 polynomials to within error $1/n$, with seed length $s = O(\log n)$,
2. a generator that fools degree-3 polynomials to within error $\epsilon$, with seed length $s = 3 \cdot \log |\mathbb{F}| n + f(\epsilon, \mathbb{F})$ where $f$ depends only on $\epsilon$ and $\mathbb{F}$,
3. assuming the “Gowers inverse conjecture,” a generator that fools degree-$d$ polynomials to within error $\epsilon$, with seed length $s = d \cdot \log |\mathbb{F}| n + f(d, \epsilon, \mathbb{F})$ where $f$ depends only on $d$, $\epsilon$, and $\mathbb{F}$.

The results in (1) and (2) are unconditional, i.e. do not rely on any unproven assumption. Moreover, the results in (3) rely on a special case of the conjecture which may be easier to prove.

Our generator for degree-$d$ polynomials is the component-wise sum of $d$ generators for degree-1 polynomials (on independent seeds).

Prior to our work, generators with logarithmic seed length were only known for degree-1 polynomials [NN90]. In fact, over small fields such as $\mathbb{F}_2 = \{0, 1\}$, our results constitute the first progress on these problems since the celebrated generator by Luby, Velicković and Wigderson [LWV93], whose seed length is much bigger: $s = \exp(\Omega(\sqrt{\log n}))$, even for the case of degree-2 polynomials over $\mathbb{F}_2$.

References


On defining integers and proving arithmetic circuit lower bounds

Peter Bürgisser

Let $\tau(n)$ denote the minimum number of arithmetic operations sufficient to build the integer $n$ from the constant 1. We prove that if there are arithmetic circuits of size polynomial in $n$ for computing the permanent of $n \times n$ matrices, then $\tau(n!)$ is polynomially bounded in $\log n$. Under the same assumption on the permanent, we conclude that the Pochhammer-Wilkinson polynomials $\prod_{k=1}^{n} (X - k)$ and the Taylor approximations $\sum_{k=0}^{n} k^{-1} X^k$ and $\sum_{k=1}^{n} k^{-1} X^k$ of $\exp$ and $\log$, respectively, can be computed by arithmetic circuits of size polynomial in $\log n$ (allowing divisions). This connects several so far unrelated conjectures in algebraic complexity.

On subexponentiality of the discrete logarithm problem in elliptic curves over extension fields

Claus Diem

Let us consider the discrete logarithm problem in the groups of rational points of elliptic curves over finite fields: Given an elliptic curve $E/\mathbb{F}_q$ and two points $P, Q \in E(\mathbb{F}_q)$ such that $Q \in \langle P \rangle$, find some $x \in \mathbb{N}$ with $Q = x \cdot P$.

We are concerned with the complexity of this algorithmic problem as a function of $\log(q)$. (Note that $\#E(\mathbb{F}_q) \sim q$, and the input length (of an appropriate representation) is in $\Theta(\log(q))$.)

The problem can obviously be solved in a running time of $\tilde{O}(q)$ by “brute-force” on a Turing machine (or on a random access machine). No (randomized) algorithm is however known which solves the problem in subexponential time. (On a random access machine the best known running time is $\tilde{O}(e^{\log(q)/2})$; this result follows from
Shanks' Baby-Step-Giant-Step algorithm which applies to the discrete logarithm problem in every finite group.

This motivates the following tasks:
Find families of elliptic curves over finite fields such that the discrete logarithm problem can be solved in an expected time which is
- subexponential, that is, in $O(q^{\epsilon})$
- bounded by a subexponentiality function, that is, bounded by a function of the form $e^{\log(q)\alpha}$ for some $\alpha < 1$.

Our result is as follows:

**Theorem 1.** There exists some $c > 0$ such that the following holds: The discrete logarithm problem in $E(\mathbb{F}_q^n)$, where $n \leq c \log(q)$ and $E$ is any elliptic curve over $\mathbb{F}_q^n$, can be solved in an expected time which is polynomial in $q$.

**Corollary 2.** Let $c$ be as above. Then the discrete logarithm problem in $E(\mathbb{F}_q^n)$, where $\frac{1}{2} \log(q)^{\epsilon} \leq n \leq \log(q)^{\epsilon}$ and $E$ is any elliptic curve over $\mathbb{F}_q^n$, can be solved in an expected time which is polynomial in $e^{\log(q)^{\frac{1}{\epsilon}+c}}$.

The corollary follows from the theorem because under the assumptions of the corollary $q = e^{\log(q)^{\epsilon} \log(q)^{\frac{1}{\epsilon}+c}} \leq e^{\frac{1}{\epsilon}n \log(q)^{\frac{1}{\epsilon}+c}}$.

Previously some families of elliptic curves for which the discrete logarithm problem is subexponential were already known. Also, using the so-called GHS attack one can prove (unpublished) that there exists a sequence of finite fields (of strictly increasing size) such that the discrete logarithm problem in all elliptic curves over these fields is subexponential. It was however not known if there exists a family of finite fields (of strictly increasing size) such that the discrete logarithm problem in all elliptic curves over these fields is bounded by a subexponentiality function. (See definitions above for the distinction of these two questions.)

The algorithm used for the proof of the theorem is essentially an algorithm given by P. Gaudry (P. Gaudry: Index calculus for abelian varieties and the elliptic curve discrete logarithm problem, preprint). As the title indicates, it is an algorithm of “index calculus” type. The relations are thereby collected by solving systems of multivariate polynomial equations over $\mathbb{F}_q$. The main difficulty of the proof of the theorem relies in analyzing the algorithm for varying extension degrees $n$.

**Extractors and Rank Extractors for Polynomial Sources**

**Zeev Dvir**

(joint work with Ariel Gabizon, Avi Wigderson)

In this work we construct explicit deterministic extractors from *polynomial sources*, namely from distributions sampled by low degree multivariate polynomials over finite fields. This naturally generalizes previous work on extraction from affine sources (which are degree 1 polynomials) [BKSSW05, Bou05, GR05]. A direct
The consequence is a deterministic extractor for distributions sampled by polynomial size arithmetic circuits over exponentially large fields.

The steps in our extractor construction, and the tools (mainly from algebraic geometry) that we use for them, are of independent interest:

The first step is a construction of rank extractors, which are polynomial mappings which "extract" the algebraic rank from any system of low degree polynomials. More precisely, for any $n$ polynomials, $k$ of which are algebraically independent, a rank extractor outputs $k$ algebraically independent polynomials of slightly higher degree. The rank extractors we construct are applicable not only over finite fields but also over fields of characteristic zero.

The next step is relating algebraic independence to min-entropy. We use a theorem of Wooley to show that these parameters are tightly connected. This allows replacing the algebraic assumption on the source (above) by the natural information theoretic one. It also shows that a rank extractor is already a high quality condenser for polynomial sources over polynomially large fields.

Finally, to turn the condensers into extractors, we employ a theorem of Bombieri, giving a character sum estimate for polynomials defined over curves. It allows extracting all the randomness (up to a multiplicative constant) from polynomial sources over exponentially large fields.

References


Counting reducible and singular bivariate polynomials

Joachim von zur Gathen

We investigate four "accidents" that can happen to a bivariate polynomial over a finite field: it can have a nontrivial factor, or a square factor, or a factor over an extension field (but none over the ground field), or a singular root, where all partial derivatives also vanish. The main results are quantitative versions of the intuition that a random polynomial is unlikely to suffer an accident.

In the set $B_n(F) \subseteq F[x,y]$ of bivariate polynomials with total degree at most $n$, we have sets $A_n(F) \subseteq B_n(F)$ of such "accidents". We have geometric and combinatorial results, namely bounds on the (minimal) codimension of $A_n$ in $B_n$ (over an algebraically closed field), or functions $\alpha_n(q)$ and $\beta_n(q)$ so that

$$\frac{\#A_n(F_q)}{\#B_n(F_q)} - \alpha_n(q) \leq \alpha_n(q) \cdot \beta_n(q).$$
The following hold for large enough $n$.

<table>
<thead>
<tr>
<th>accident</th>
<th>codim</th>
<th>$\alpha_n(q)$</th>
<th>$\beta_n(q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>reducible</td>
<td>$n - 1$</td>
<td>$(q + 1)q^{-n}$</td>
<td>$2q^{-n+3}$</td>
</tr>
<tr>
<td>squareful</td>
<td>$2n - 1$</td>
<td>$\frac{q^{-2n+1} + (1+q^{-1}) + q^{-2n-1}}{1-q^{-2n+1}}$</td>
<td>$6q^{-2n+6}$</td>
</tr>
<tr>
<td>rel. irreducible</td>
<td>$\varepsilon_n(q)$</td>
<td>$2q^{-n+d+1}$</td>
<td></td>
</tr>
<tr>
<td>singular</td>
<td>1</td>
<td>$\frac{(1 - q^{-3})^2}{l}$</td>
<td>0</td>
</tr>
</tbody>
</table>

In the third line, we use $l$ for the largest prime divisor of $n$, and

$$\varepsilon_n(q) = \frac{q^{-n+1} - 1}{1 - q^{-1}}.$$ 


**REFERENCES**


**On the Approximation Resistance of a Random Predicate**

**Johan Håstad**

Consider a predicate $P$ which takes as input $k$ Boolean variables and outputs true/false. Suppose $P$ accepts $t_P$ of the $2^k$ possible input strings.

For the Max-CSP connected with $P$ an instance is given by a $k$-tuple of literals. For each assignment we can observe the number of $k$-tuples of Boolean variables that satisfy $P$ and the goal is to maximize this number.

This problem is NP-hard for almost all $P$ and we look at algorithms that approximate this number. There is a natural algorithm that approximates this number
Complexity Theory

within $t_P 2^{-k}$ which simply picks a random assignment. We say that a predicate is approximation resistant if it is hard to get a significantly stronger approximation guarantee. To be more precise $P$ is approximation resistant if, for any $\epsilon > 0$, it is NP-hard to approximate the maximal number of simultaneously satisfiable constraints within a factor $t_P 2^{-k} + \epsilon$.

We prove that, assuming the unique games conjecture [1], for sufficiently large $k$ a random predicate is approximation resistant with high probability.

The result builds on a recent result by Samorodnitsky and Trevisan [2] that if $2^d$ is the smallest power of two larger than $k$, there is predicate $P_{ST}$ of width $k$ that only accepts $2^d$ strings and is approximation resistant.

Our proof shows that any predicate implied by $P_{ST}$, or a predicate obtained from $P_{ST}$ by permuting the inputs and negating some input bits also is approximation resistant. Using a second moment method we then show that this criteria applies to a random predicate.

This paper will be published at the Approx07-conference to be held in August 2007.

References


The Black-Box Query Complexity of Polynomial Summation

Valentine Kabanets

(joint work with Ali Juma, Charles Rackoff, Amir Shpilka)

For any given Boolean formula $\phi(x_1, \ldots, x_n)$, one can efficiently construct (using arithmetization) a low-degree polynomial $p(x_1, \ldots, x_n)$ that agrees with $\phi$ over all points in the Boolean cube $\{0, 1\}^n$; the constructed polynomial $p$ can be interpreted as a polynomial over an arbitrary field $\mathbb{F}$. The problem $#SAT$ (of counting the number of satisfying assignments of $\phi$) thus reduces to the polynomial summation $\sum_{x \in \{0, 1\}^n} p(x)$. Motivated by this connection, we study the query complexity of the polynomial summation problem: Given (oracle access to) a polynomial $p(x_1, \ldots, x_n)$, compute $\sum_{x \in \{0, 1\}^n} p(x)$. Obviously, querying $p$ at all $2^n$ points in $\{0, 1\}^n$ suffices. Is there a field $\mathbb{F}$ such that, for every polynomial $p \in \mathbb{F}[x_1, \ldots, x_n]$, the sum $\sum_{x \in \{0, 1\}^n} p(x)$ can be computed using fewer than $2^n$ queries from $\mathbb{F}^n$? We show that the simple upper bound $2^n$ is in fact tight in the black-box model where one has only oracle access to the polynomial $p$, for any field $\mathbb{F}$. We prove these lower bounds for the adaptive query model, where the next query can depend on the values of $p$ at previously queried points. Our lower bounds hold even for polynomials that have degree at most 2 in each variable. In contrast, for polynomials
that have degree at most 1 in each variable (i.e., multilinear polynomials), we show that a single query is sufficient over any field of characteristic other than 2.

On probabilistic analysis of randomization in hybrid symbolic-numeric algorithms
Erich Kaltofen
(joint work with Zhengfeng Yang, Lihong Zhi)

Algebraic randomization techniques can be applied to hybrid symbolic-numeric algorithms, that is, algebraic algorithms where the scalars in the inputs have numerical errors. We consider the problem of solving highly over- and underdetermined systems of linear equations by essentially optimal randomized algorithms (e.g., solving a linear system with \( n \) equations and \( p = O((n \log(n))^{0.72}) \) variables in \( O(pn \log(n)) \) field operations) and interpolating a sparse rational multivariate function from noisy values. We show that Zippel's original sparse polynomial interpolation technique applies to numerically perturbed data and we give an exact and hybrid algorithm for interpolating sparse rational functions. We discuss the expected condition numbers of the arising randomized linear systems, and observe that certain randomized projections can lead to ill-conditioned systems [1].

References

Understanding parallel repetition requires understanding foams
Guy Kindler
(joint work with Uri Feige, Ryan O'Donnell)

The parallel repetition theorem, proven by Raz in 1995, is a fundamental result that in addition to its philosophical appeal, plays a key role in complexity theory. The theorem studies the behavior of success probabilities of two prover games, when many copies of such a game are played in parallel. It shows that the success probability decreases exponentially in the number of repetitions, but the parameters given by the theorem do not seem tight. It is natural to ask what are the best parameters for which the theorem holds, and improving them would have complexity theoretic implications.

This talk describes an attempt to improve the parameters in a very special case of the parallel repetition theorem. Our attempt had only limited success, but it turns out that the reason we got stuck was that the following seemingly hard question from the geometry of "foams" was hidden in the special case that we
were trying to solve: What is the least surface area of a cell that tiles $\mathbb{R}^d$ by the lattice $\mathbb{Z}^d$? Very little about this foam problem is known. It is interesting to see such a geometric question encoded inside the problem of parallel repetition in two prover games.

**Interpolation in Valiant’s theory**

**Pascal Koiran**

(joint work with Sylvain Périfel)

The starting point of this work is a question raised by Christos Papadimitriou in a personal communication to Erich Kaltofen:

**Question (**)**

If a multivariate polynomial $P$ is computable by a (boolean) polynomial-time algorithm on rational inputs, does that imply that $P$ can be computed by a polynomial-size arithmetic circuit? In such a circuit, the only allowed operations are additions, subtractions, and multiplications.

As pointed out by Papadimitriou, Strassen’s "Vermeidung von Divisionen" shows that that for evaluating a low-degree polynomial $P$, divisions would not increase exponentially the power of arithmetic circuits. It is indeed a natural question whether, more generally, all boolean operations can be replaced efficiently by additions, subtractions and multiplications.

In my talk I explained why it seems difficult to answer this question either way. Obtaining a positive answer seems difficult because it would imply the following transfer theorem: $\text{FP} = \sharp P \Rightarrow \text{VP} = \text{VNP}$ (assuming that $\text{FP} = \sharp P$, the permanent must be in $\text{FP}$; a positive answer to question (**) would therefore imply that the permanent is in $\text{VP}$, and that $\text{VP} = \text{VNP}$ by completeness of the permanent).

Unfortunately, in spite of all the work establishing close connections between the boolean model of computation and the algebraic models of Valiant and of Blum, Shub and Smale no such transfer theorem is known. In fact, we do not know of any hypothesis from boolean complexity theory that would imply the equality $\text{VP} = \text{VNP}$ (but transfer theorems in the opposite direction were established [1]).

A natural strategy for obtaining a negative answer to question (**) would be to exhibit a family of polynomials that are easy to evaluate on rational inputs but hard to evaluate by arithmetic circuits. Unfortunately, there seems to be a lack of candidate polynomials. Another difficulty is that a negative answer would imply the separation of the algebraic complexity classes $\text{VP}^0$ and $\text{VNP}^0$. This observation is our main contribution to the study of question (**) The classes $\text{VP}^0$ and $\text{VNP}^0$ are constant-free versions of the classes $\text{VP}$ (of "easily computable polynomial families") and $\text{VNP}$ (of "easily definable polynomial families") defined by Valiant. The separation $\text{VP}^0 \neq \text{VNP}^0$ seems very plausible, but it also seems very difficult to establish.

The full paper will contain a few additional results.
Subspace polynomials and List Decoding of Reed-Solomon Codes

SWASTIK KOPPARTY
(joint work with Eli Ben-Sasson, Jaikumar Radhakrishnan)

We show combinatorial limitations on efficient list decoding of Reed-Solomon codes beyond the Johnson and Guruswami-Sudan bounds [Joh62, GS99]. In particular, we show that for arbitrarily large fields $\mathbb{F}_N, |\mathbb{F}_N| = N$, for any $\delta \in (0, 1)$, and $K = N^{\delta}$:

- **Existence:** there exists a received word $w_N : \mathbb{F}_N \rightarrow \mathbb{F}_N$ that agrees with a super-polynomial number of distinct degree $K$ polynomials on $\approx N^{1/\delta}$ points each;
- **Explicit:** there exists a polynomial time constructible received word $w'_N : \mathbb{F}_N \rightarrow \mathbb{F}_N$ that agrees with a super-polynomial number of distinct degree $K$ polynomials, on $\approx 2^{\sqrt{KN}}$ points each.

In both cases, our results improve upon the previous state of the art, which was $\approx N^{1/\delta}$ points of agreement for the existence case [JH01], and $\approx 2N^{1/2}$ points of agreement for the explicit one [GR05b]. Furthermore, for $\delta$ close to 1 our bound approaches the Guruswami-Sudan bound (which is $\sqrt{NK}$) and implies limitations on extending their efficient Reed-Solomon list decoding algorithm to larger decoding radius.

Our proof method is surprisingly simple. We work with polynomials that vanish on subspaces of an extension field viewed as a vector space over the base field. These *subspace polynomials* are a subclass of *linearized polynomials* that were first studied by Ore [Ore33, Ore34] in the 1930s, and later by coding theorists. For us their main attraction is their sparsity and abundance of roots, virtues that recently won them pivotal roles in *probabilistically checkable proofs of proximity* [BSGH+04, BSS05] and sub-linear proof verification [BSGH+05].

References


We construct boolean functions $f : \{0,1\}^n \rightarrow \{0,1\}^m$, for which the graph $g^r(f) = \{(x, f(x)), x \in \{0,1\}\} \subseteq \{0,1\}^{n+m}$ has maximal algebraic immunity. Hereby, the algebraic immunity $AI(S)$ of a subset $S$ of the boolean cube is defined to be the minimal $d$ for which there is a nontrivial degree-$d$ polynomial (over $\mathbb{F}_2$) which annihilates $S$, i.e., which outputs 0 for all $x \in S$. Consequently, if the algebraic immunity of a given boolean function is $d$ then nontrivial relations in the input/output bits of degree smaller than $d$ do not exist.

The study of the algebraic immunity of boolean functions in the context of symmetric cryptography was initiated by Meier, Pasalic, Carlet in [10]. It is motivated by the need for appropriate boolean functions serving as building blocks of symmetric ciphers. Such functions should have large algebraic immunity for preventing vulnerability of the cipher against algebraic attacks. For several practically used cryptosystems, building blocks with low algebraic immunity open the door to express the bits of the secret key by an overdetermined system of low degree equations (see, e.g., [2], [6], [7], [8], [13]), which then can be solved by nontrivial methods ([4], [5], [12]).

In [3] we completely solve the problem of constructing explicitly defined single-output functions of maximal algebraic immunity. For even number of input bits it can be easily shown that majority has this property. For odd number of input bits the situation is more complicated.

For multi-output functions no explicit construction of a function family of maximal algebraic immunity is known. We present an efficient algorithm, based on the matroid union algorithm of Edmonds [9], which computes for given $m, n, d$, $\ldots$
if existent, the table of a function \( h : \{0, 1\}^n \rightarrow \{0, 1\}^m \) of algebraic immunity \( d \).

To the best of our knowledge, this is the first systematic method for constructing multi-output functions of high algebraic immunity.

A natural upper bound \( d'(n, m) \) for the algebraic immunity of a boolean function \( f : \{0, 1\}^n \rightarrow \{0, 1\}^m \) is the minimal number \( d \) such that \( \sum_{i=0}^{d} \binom{n+m}{i} > 2^n \).

We conjecture that for all \( 1 \leq m \leq n \) there are functions \( f : \{0, 1\}^n \rightarrow \{0, 1\}^m \) of algebraic immunity \( d'(n, m) \). Experiments show that this is true for all \( 1 \leq m \leq n \leq 20 \) [1]. The proof of this conjecture remains as an open problem.

References

All Natural NPC Problems Have Average-Case Complete Versions  
NOAM LIVNE

In 1984 Levin put forward a suggestion for a theory of *average case complexity* [1]. In this theory a problem, called a *distributional problem*, is defined as a pair consisting of a decision problem and a probability distribution over the instances. Introducing adequate notions of "efficiency-on-average", simple distributions and efficiency-on-average preserving reductions, Levin developed a theory analogous to the theory of NP-completeness. In particular, he showed that there exists a simple distributional problem that is complete under these reductions. But since then very few distributional problems were shown to be complete in this sense. In this paper we show a simple sufficient condition for an NP-complete decision
problem to have a distributional version that is complete under these reductions (and thus to be “hard on the average” with respect to some simple probability distribution). Apparently all known NP-complete decision problems meet this condition.

**References**


**The probability that a small perturbation of a numerical analysis problem is difficult**

**Martin Lotz**

(joint work with Peter Bürgisser, Felipe Cucker)

The *condition number* of a numerical computation problem measures the sensitivity of the output to small perturbations of the input. Condition numbers occur in many instances of round-off analysis, and they also appear as a parameter in complexity bounds for a variety of iterative algorithms for solving numerical problems. In the work underlying this talk [1], we prove a general result providing smoothed analysis estimates for conic condition numbers. Our probability estimates depend only on geometric invariants of the corresponding sets of ill-posed inputs.

A condition number is *conic* if there exists a semi-algebraic cone $\Sigma \subseteq \mathbb{R}^{p+1}$, the set of ill-posed inputs, such that for all input data $a \in \mathbb{R}^{p+1} \setminus \{0\}$, $|a| = \frac{1}{\text{dist}(a, \Sigma)}$ holds. Since $\Sigma$ is a cone, we may restrict to data lying in the unit sphere $S^p$, and then the conic condition number be characterized as the inverse distance to ill-posedness on the sphere. Our main result is the following (in the statement, $z \in B(a, \sigma)$ means that $z$ is uniformly distributed in a spherical cap of radius $\arcsin \sigma$ around $a$).

**Theorem 1.** Let $\sigma$ be a conic condition number with set of ill-posed inputs $\Sigma$, and assume $\Sigma$ is contained in the zero set of homogeneous polynomials of degree at most $d$. Then, for all $\sigma \in (0, 1]$ and all $t \geq (2d + 1) \frac{1}{\sigma}$,

$$\sup_{a \in S^p} \text{Prob}_{z \in B(a, \sigma)} \{ |z| \geq t \} \leq 26 dp \frac{1}{\sigma t},$$

and

$$\sup_{a \in S^p} E_{z \in B(a, \sigma)} (\ln |z|) \leq 2 \ln p + 2 \ln d + 2 \ln \frac{1}{\sigma} + 5.$$

While many condition numbers are not conic themselves, they can often be bounded by such. Several applications to linear and polynomial equation solving show that the estimates obtained in this way are easy to derive and quite accurate.

The main theorem is based on a volume estimate of $\varepsilon$-tubular neighborhoods around a real algebraic subvariety of a sphere, intersected with a disk of radius $\sigma$. Besides $\varepsilon$ and $\sigma$, this bound depends only the dimension of the sphere and on the degree of the defining equations.
Combinatorial Construction of Locally Testable Codes

Or Meir

An error correcting code is said to be locally testable if there is a test that can check whether a given string is a codeword of the code, or rather far from the code, by reading only a constant number of symbols of the string. Locally Testable Codes (LTCs) were first explicitly studied by Goldreich and Sudan [4] and since then few constructions of LTCs were suggested (see [3] for a survey of those constructions).

LTCs are connected with Probabilistically Checkable Proofs (PCPs) and can be seen as the "Combinatorial counterparts" of PCPs. Since they are simpler objects then PCPs, one might expect that constructing LTCs would be easier than constructing PCPs. However, all the known constructions either use PCP as a building block, or imply directly the existence of a PCP.

In this work we present a new and simpler construction of LTCs that seems to be strictly weaker than PCP. Another important feature of our construction is that it is purely combinatorial, while previous constructions were very algebraic. Finally, our construction matches the parameters of the best known construction of LTCs by Ben-Sasson and Sudan [1] (in both cases, these constructions are combined with Dinur’s gap amplification technique [2] in order to achieve the best possible parameters). However, unlike the construction of [1], our construction is not entirely explicit.

REFERENCES


Sub-Constant Error Low Degree Test and PCP of Almost-Linear Size

Dana Moshkovitz

(joint work with Ran Raz)

The PCP theorem [2, 1] is one of the most important theorems proven in Theoretical Computer Science. The PCP theorem states that any mathematical proof can be written in a different format, such that the proof can be (probabilistically)
verified by querying only a constant number of places in it. The PCP theorem implies hardness of approximation problems, as well as yields constructions of codes with local testing and decoding properties. Two parameters of a PCP that play a central role are the size of the PCP and its probability of error.

In 1997 researchers managed to construct PCPs with polynomial size and sub-constant probability of error [11, 3, 7]. In the last 6 years, many researchers got extremely interested in PCPs of almost linear size and managed to construct such PCPs [8, 5, 4, 6]. However, these last constructions of PCPs with almost linear size only achieve constant (and not sub-constant) probability of error. The bottleneck for constructing PCPs that have both sub-constant error and almost linear size was the construction of low degree tests that have both sub-constant error and almost linear size. We constructed such tests [9] and proved a corresponding PCP theorem [10].

References


The Cryptographic Applications of Compressibility With Respect to Solutions

MONI NAOR

(joint work with Danny Harnik)

We study compression that preserves the solution to an instance of a problem rather than preserving the instance itself. Our focus is on the compressibility of NP decision problems. We consider NP problems that have long instances but relatively short witnesses. The question is, can one efficiently compress an instance and store a shorter representation that maintains the information of whether the
original input is in the language or not. We want the length of the compressed instance to be polynomial in the length of the witness and polylogarithmic in the length of original input. We discuss the differences between this notion and similar notions from parameterized complexity.

Our motivation for studying this issue stems from the vast cryptographic implications such compressibility has. For example, we say that SAT is compressible if there exists a polynomial $p$, so that given a formula consisting of $m$ clauses over $n$ variables it is possible to come up with an equivalent (w.r.t satisfiability) formula of size at most $p(n, \log m)$. Then, given a compression algorithm for SAT we provide a construction of: (i) A one-way function from a distributionally-hard problem. (ii) Collision resistant hash functions from any one-way function. The latter task was shown to be impossible via black-box reductions by Simon [4], and indeed the construction presented is inherently non-black-box. Another application of SAT compressibility is a cryptanalytic result concerning the limitation of everlasting security in the bounded storage model (see [1, 2]) when mixed with (time) complexity based cryptography.

References


Two problems related to the Max-Cut and the Unique Games Conjecture

Ryan O’Donnell

This talk is in two parts.

In the first part we report on recent work with Yi Wu [1]. In this work, we complete a long line of research into SDP algorithms and hardness results for the Max-Cut problem. Specifically, we explicitly identify a certain curve $S : [\frac{1}{2}, 1] \rightarrow [\frac{1}{2}, 1]$ with the following properties: For each $c \in [\frac{1}{2}, 1]$, there is a graph with Max-Cut at most $S(c)$ but SDP relaxation at least $c$. On the other hand, every graph with SDP relaxation at least $c$ has a cut of value at least $S(c)$, and further, this cut is findable via an efficient “RPR2” SDP algorithm. Furthermore, we connect SDP analysis to Long Code test analysis and show the following: Among all (Max-Cut) Long Code tests with completeness at least $c$, the lowest possible achievable soundness is $S(c)$. Further consequences of these results for algorithmic hardness...
In the second part of the talk, we discuss a certain aspect of the Unique Games Conjecture we feel is overlooked. Namely, we do not know any distribution on Unique-Label-Cover instances — natural or not — for which it seems harder to approximate solutions better than the extent for which we know NP-hardness. Or more concretely, we do not know a distribution on Max-2Lin(2) instances with value 1 − ε for which finding 1 − Ω(ε) solutions even “seems” hard — whereas the Unique Games Conjecture predicts that even finding 1 − Ω(√ε) solutions should be hard. This is in contrast to, say, Max-3Lin(2), where for the most natural random planted 1 − ε instances, finding 1 + ε empirically seems very hard. We propose as an open problem looking for distributions on Max-2Lin(2) or Max-2Lin(q) instances that seem hard to approximate.

References


Tight Integrality gaps for Vertex Cover SDPs in the Lovasz-Schrijver hierarchy

TONIANN PITASSI

(joint work with Konstantinos Georgiou, Avner Magen, Iannis Tourlakis)

Linear and semidefinite programming are highly successful approaches for obtaining good approximations for NP-hard optimization problems. For example, breakthrough approximation algorithms for Max Cut and Sparsest Cut use semidefinite programming.

Perhaps the most prominent NP-hard problem whose exact approximation factor is still unresolved is Vertex Cover. PCP-based techniques of Dinur and Safra show that it is not possible to achieve a factor better than 1.36; on the other hand no known algorithm does better than the factor of 2 achieved by the simple greedy algorithm. Furthermore, there is a widespread belief that SDP techniques are the most promising methods available for improving upon this factor of 2.

Following a line of study initiated by Arora, Bollobas, Lovasz and Tourlakis, our aim is to show that a large family of LP and SDP based algorithms fail to produce an approximation for Vertex Cover better than 2. Lovasz and Schrijver introduced the LS systems that naturally capture large classes of LP and SDP relaxations. The strongest of these systems, $LS_+$, captures the celebrated SDP-based algorithms for Max Cut and Sparsest Cut mentioned above.

We prove an integrality gap of 2 for Vertex Cover SDPs resulting from tightening the standard LP relaxation with $\Omega(\sqrt{\log n / \log \log n})$ rounds of $LS_+$. While tight integrality gaps for Vertex Cover were known for the weaker $LS$ system previous results did not preclude a polynomial-time $2-\Omega(1)$ approximation algorithm based on $LS_+$, even when restricted to only two rounds of $LS_+$ tightenings.
Quantum Frege proofs and a problem on quantum computing

Pavel Pudlák

In this talk I shall address the question whether quantum circuits could help us prove theorems of the classical propositional calculus faster than conventional devices. I shall propose a class of proof systems, *Quantum Frege Proof Systems*. This is based on a generalization of the concept of a Frege deduction rule to the quantum setting. A *quantum Frege rule* is roughly a linear superposition of classical Frege rules. Given a finite set of quantum deduction rules, a *quantum Frege proof* is a sequence of proof lines, the first line being empty and each next line is obtained from the previous one by applying one of the quantum deduction rules. Thus each proof line is a quantum superposition of strings of formulas. We say that the proof proves a given proposition $\phi$ if $\phi$ occurs in the last proof line with amplitude $|\alpha| \geq 1/2$ (i.e., if we measure the last state we shall see a string of propositions which includes $\phi$ with probability $\geq 1/2$). We represent a quantum Frege proof by a string of quantum circuits that compute the transitions defined by the quantum Frege rules.

Given a quantum Frege proof $P$ of a tautology $\phi$, one can easily show that there exists a classical Frege proof with the same number of steps and the same bound on the size of formulas involved. However, if we are given a representation of $P$ by the string of quantum circuits, we do not know how to construct this classical proof. We can show that a classical proof cannot be constructed in polynomial time, if factoring is not computable in polynomial time. The proof of this result is based on tautologies formalizing a bit commitment schema. What remains open is whether one can construct the classical proof using polynomial size quantum circuits. This is closely related to the following problem about histories.

Let $B$ be the basis of the Hilbert space of $n$ qubits consisting of the $2^n$ strings of 0-1 bits. Let $K = (U_1, \ldots, U_t)$ be a string of unitary operators. For $a_0, a_1, \ldots, a_t \in B$, we shall say that $(a_0, a_1, \ldots, a_t)$ is a history of $K$, if for all $i = 1, \ldots, t$, $\langle a_i | U_i | a_{i-1} \rangle \neq 0$. Let $U = U_t \cdots U_1$ denote the product of a string of the unitary transformations. If $a, b \in B$ are such that $\langle b | U | a \rangle \neq 0$, then there exists a history of the form $(a = a_0, a_1, \ldots, a_t = b)$.

**Problem.** Suppose the unitary transformations $U_i$ are given by quantum circuits $C_i$. Let also $a \in B$ be given and assume that measuring the first bit of $|U|a\rangle$ gives 0 with probability at least $1/2$. Is it possible to construct a history $(a = a_0, a_1, \ldots, a_t)$ such that the first bit of $a_t$ is 0 using polynomial size quantum circuits?

Related questions have been studied in [1].

**References**


Complexity Theory

Randomness versus Hardness and Lower Bounds for Constant-Depth Arithmetic Circuits

Ran Raz

I gave a short description of the main results in [R].

We present simple-to-describe problems, that seem natural-to-study in the context of pseudorandomness and explicit constructions of combinatorial objects, and are seemingly unrelated to arithmetic circuit complexity, and whose solution would give strong (up to exponential) lower bounds for the size of general arithmetic circuits. We then prove lower bounds of \( n^{1+O(1/d)} \) for the size of arithmetic circuits of depth \( d \) for explicit polynomials of degree \( O(d) \).

Our main results are as follows: Let \( \mathbb{F} \) be a field and let \( n \) be an integer.

1. Let \( s = s(n), m = m(n), r = r(n) \) be integers s.t. \( n^{-1} \leq s < m = n^r \).

   (Think of \( r \) as relatively small, say \( r = \log \log n \).

   Can one give an explicit polynomial-mapping \( f : \mathbb{F}^n \to \mathbb{F}^m \) of total-degree at most \( 2^n \), such that, the image of \( f \) is not contained in the image of any polynomial-mapping \( \Gamma : \mathbb{F}^s \to \mathbb{F}^m \) of total-degree at most \( r \)?

   We show that for any \( \mathbb{F} \) of characteristic different than 2, and any \( s, m, r \) as above, the existence of an explicit \( f \) as above (with the right notion of explicitness) implies super-polynomial lower bounds for computing the permanent over \( \mathbb{F} \).

2. Let \( s = s(n), m = m(n), r \) be integers s.t. \( s < m = n^{r+1} \), and \( 2 \leq r \leq O(1) \).

   Given (as input) a polynomial-mapping \( \Gamma : \mathbb{F}^s \to \mathbb{F}^m \) of total-degree at most \( 2^r - 1 \), can one output (in polynomial time) an explicit polynomial-mapping \( f : \mathbb{F}^n \to \mathbb{F}^m \) of total-degree at most \( \text{poly}(n) \), such that, the image of \( f \) is not contained in the image of \( \Gamma \)?

   We show that for any \( \mathbb{F} \) and any \( s, m, r \) as above, a positive solution for this problem implies an explicit lower bound of \( \Omega(\sqrt{s}) \) for the size of arithmetic circuits over \( \mathbb{F} \).

3. For any \( d = d(n) \), we give an explicit example for an \( n \)-variate polynomial of total-degree \( O(d) \), with coefficients in \( \{0,1\} \), such that, any depth \( d \) arithmetic circuit for this polynomial (over any field) is of size \( \geq n^{1+O(1/d)} \).

REFERENCES


Flag Algebras and Density of Triangles in Graphs

Alexander Razborov

This talk is devoted to the part of Extremal Combinatorics that, in the asymptotic form, studies with which densities “template” combinatorial structures (like graphs, digraphs, hypergraphs or tournaments) may or may not appear in unknown (large) structures of the same type. It is worth noting that the whole subject of


Extremal Combinatorics originated in the seminal paper by Turán (1941) devoted to problems of exactly this type (this is why the densities in question are also called Turán densities). And, although by now the subject has definitely outgrown these boundaries, it would be fair to say that Turán-like problems still make its core.

Consider for example three problems of a very similar flavour. What is the minimal edge density of a graph that guarantees the existence of at least one copy of $K_3$ in this graph? Supposing the edge density is greater than this critical value, what is the minimal possible density of triangles guaranteed to exist in such a graph (as a function of the edge density)? How about the analogous questions for 3-hypergraphs (with $K_3$ replaced by $K_4^3$, the complete 3-graph on 4 vertices)?

Of these three questions, the first one was completely solved in the seminal paper by Turán (in fact, for more general case of $K_r$, where $r \geq 3$ is an arbitrary constant). The second question is answered in our work, but its generalization to $K_r$ is still open for any $r > 3$. The third question is widely open: this is one of the most intriguing, famous and notoriously difficult problems in the whole field of Combinatorics.

Except for the above-mentioned concrete result, we also try to understand and explicitly extract the mathematical structure underlying and unifying many common techniques existing in the “asymptotic” Extremal Combinatorics and find for them a common denominator. The backbone of our framework is made by certain associative commutative algebras over the reals that we call “Flag Algebras”; most of the standard ideas in the area can be then expressed as simple computations in these algebras using a small set of pre-defined homomorphisms and linear mappings between them. This framework captures, among other things, various inductive arguments existing in the area, and, after some routine technical work (done once and for all) it becomes completely free of the $\epsilon/\delta$ stuff. In this sense it can be viewed as an extremely goal-oriented fragment of the non-standard analysis; another related feature is that computations in the flag algebras are very easy to program, which substantially enhances the search for right relations and techniques useful for any given concrete problem.

The talk is based on two preprints: “Flag Algebras” (to appear in Journal of Symbolic Logic) and “On the Minimal Density of Triangles in Graphs” (to appear in Combinatorics, Probability and Computing); both are available from my homepage: http://www.mi.ras.ru/~razborov/.

A Hypercontractive Inequality for Matrix-Valued Functions

Oded Regev

(joint work with Avraham Ben-Aroya and Ronald de Wolf)

The Bonami-Beckner hypercontractive inequality is a powerful tool in Fourier analysis of real-valued functions on the Boolean cube. We present a version of this inequality for matrix-valued functions on the Boolean cube. We also present a number of applications of this. In particular, we analyze maps that encode $n$
classical bits into $m$ qubits, in such a way that each set of $k$ bits can be recovered with some probability by an appropriate measurement on the quantum encoding; we show that if $m < 0.7n$, then the success probability is exponentially small in $k$. This may be viewed as a direct product version of Nayak’s quantum random access code bound. It in turn implies strong direct product theorems for the one-way quantum communication complexity of Disjointness and other problems. We also slightly strengthen and simplify a result about 3-party communication complexity of Disjointness due to Beame et al.

References


Partial Exposure and Correlated Types in Large Games

OMER REINGOLD

(joint work with Ronen Gradwohl)

In this work we introduce the notion of partial exposure, in which the players of a simultaneous-move Bayesian game are exposed to the realized types and chosen actions of a subset of the other players. We show that in any large simultaneous-move game, each player has very little regret even after being partially exposed to other players. Additionally, in any extensive version of the game in which a player is partially exposed to other players, her original strategy is very likely still a best response.

Furthermore, we generalize the recent results of Kalai (2004, 2005) [3, 4], and show that the equilibria of large continuous games with many semi-anonymous players are ex post Nash and structurally robust even when the types are correlated. Two forms of correlation are permitted: local dependencies, in which each player’s type can depend arbitrarily on some fixed set of other players, and “peer-pressure” dependencies, in which any set of $k$ or more players may be mutually dependent (but any $k-1$ are independent).

Finally, we combine the above and show a robustness result for all large games, even with correlated types.
In the talk we aimed to discuss a central notion to our work, which is the effect of random variables on a function [1, 2]. We aimed at discussing the similarity and differences between the effect and the influence of random variables.

REFERENCES

Designing Boolean Sorting Circuits with Optimal Average Delay
RÜDIGER REISCHUK
(joint work with A. Jakoby, M. Liškiewicz, C. Schindelhauer)

In previous work we have introduced an average case measure for the time complexity of Boolean circuits - that is the delay between feeding the input bits into a circuit and the moment when the results are ready at the output gates - and analysed this complexity measure for prefix computations. Here we consider the problem to sort large integers that are given in binary notation. Contrary to a word comparator sorting circuit $C$ where a basic computational element, a comparator, is charged with a single time step to compare two elements, in a bit comparator circuit $C'$ a comparison of two binary numbers has to be implemented by a Boolean subcircuit $CM$ called comparator module that is built from Boolean gates of bounded fanin. Thus, compared to $C$, the depth of $C'$ will be larger by a factor up to the depth of $CM$.

Our goal is to minimize the average delay of bit comparator sorting circuits. The worst-case delay can be estimated by the depth of the circuit. For this worst-case measure two topologically quite different designs seem to be appropriate for the comparator modules: a tree-like one if the inputs are long numbers, otherwise a linear array working in a pipelined fashion. Inserting these into a word comparator circuit we get bit level sorting circuits for binary numbers of length $m$ for which the depth is either increased by a multiplicative factor of order $\log m$ or by an additive term of order $m$.

We show that this obvious solution can be improved significantly by constructing efficient sorting and merging circuits for the bit model that only suffer a constant factor time loss on the average if the inputs are uniformly distributed. This is done by designing suitable hybrid architectures of tree compaction and pipelining. These results can also be extended to classes of nonuniform distributions if we put a bound on the complexity of the distributions themselves.
Complexity Theory

57

REFERENCES


Some Specific Derandomizations

NITIN SAXENA

1. Towards Depth 3 Identity Testing and Lower Bounds

We study depth-3 arithmetic circuits ($\Sigma \Pi \Sigma$ circuits) of the form:

$$C(x_1, \ldots, x_n) = \sum_{i=1}^{k} \ell_{i,1}^{e_{i,1}} \cdots \ell_{i,c}^{e_{i,c}}$$

where, $\ell_{i,1}, \ldots, \ell_{i,c}$ are linear functions over a field $F$, $c$ is a constant and say $(e_{i,1} + \cdots + e_{i,c}) =: d$ which is the total degree of the polynomial $C(x_1, \ldots, x_n)$. We show that identity testing of such circuits can be done in $\text{poly} \left( \max_i \{ (e_{i,1} + 1) \cdots (e_{i,c} + 1) \} \right)$, $k, n$ many field operations. This immediately gives a $\text{poly}(2^d, k, n)$ time identity test for general depth 3 circuits. We also show exponential lower bounds for determinant and permanent for circuits of the above form. Our lower bounds hold over any field $F$. Proving such lower bounds for general depth 3 circuits over fields of characteristic 0 is an important open problem [2].

2. Towards Polynomial Factoring over finite fields (assuming GRH)

Finding a nontrivial factor of a given univariate polynomial over a finite field is a fundamental algebraic problem. It has a randomized polynomial time algorithm but its deterministic complexity is open. There are many partial results known using the Generalized Riemann Hypothesis, see [1] for references. We extend all these approaches and relate the problem of polynomial factoring (assuming GRH) with the existence of some combinatorial objects called association schemes. We conjecture that certain anti-symmetric association schemes do not exist which would then imply that polynomial factoring is in P (assuming GRH).
On the Complexity of Counting Components of Algebraic Varieties
Peter Scheiblechner
(joint work with Peter Bürgisser)

We consider complex algebraic varieties $V = Z(f_1, \ldots, f_r) \subseteq \mathbb{C}^n$ given by finitely many polynomials $f_1, \ldots, f_r \in \mathbb{C}[X_1, \ldots, X_n]$. A standard argument shows that the complexity of the following problems does not essentially change when changing the input data structure from dense to sparse or straight-line program representation.

For simplicity we state our results in the Turing model only, where we restrict ourselves to rational polynomials. We consider the following problems:

- $\#CC$: Given $f_1, \ldots, f_r$, compute the number of connected components of $V$.
- $\#IC$: Given $f_1, \ldots, f_r$, compute the number of irreducible components of $V$.
- $\#Betti(k)$: Given $f_1, \ldots, f_r$, compute the $k$th topological Betti number of $V$.

Furthermore, we denote by $\#IC(r)$ the problem $\#IC$ restricted to a fixed number $r$ of equations. Our main results are summarised in the following table.

<table>
<thead>
<tr>
<th></th>
<th>$#Betti(k)$</th>
<th>$#CC$</th>
<th>$#IC$</th>
<th>$#IC(r)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSPACE</td>
<td>hard</td>
<td>complete</td>
<td>contained</td>
<td></td>
</tr>
<tr>
<td>$#P$</td>
<td></td>
<td></td>
<td></td>
<td>hard</td>
</tr>
<tr>
<td>$P$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random NC</td>
<td></td>
<td></td>
<td></td>
<td>contained</td>
</tr>
</tbody>
</table>

We note that in the algebraic model one can derandomise the result for $\#IC(r)$ at the cost of good parallelisation, i.e., the problem can be solved in deterministic sequential polynomial time in the algebraic model.

References


Cryptography Based on the Equivalence of Quadratic Forms

CLAUS PETER SCHNORR
(joint work with R.J. Hartung)

Notation. A symmetric matrix $A = A^t \in \mathbb{Z}^{n \times n}$ defines the quadratic form $\bar{x}^t A \bar{x}$. The forms $A_0, A_1 \in \mathbb{Z}^{n \times n}$ are equivalent if $T^t A_0 T = A_1$ holds for some $T \in \text{GL}_n(\mathbb{Z})$.

References. [Ca78] presents the classical theory of rational quadratic forms, for LLL-reduction of quadratic forms see [S07, Si05] and for lattice based cryptography see [MG02].

We present public key identification and digital signatures based on the computational equivalence problem (CEP) of $n$-ary quadratic forms, $n \geq 3$. We present proofs of knowledge of an equivalence transform $T \in \text{GL}_n(\mathbb{Z})$. Small dimension $n$ yields short private and public keys and efficient protocols.

Lattices correspond to positive definite quadratic forms. However, lattice based cryptography requires lattices of high dimension $n$ because the lattice problems SVP and CVP are in polynomial time for any fixed dimension $n$ and get slowly harder as $n$ increases. Importantly, solving CEP for a small $T$ such that $T^t A_1 T = A_0$ is NP-hard for indefinite forms $A_1, A_0 \in \mathbb{Z}^{n \times n}$ for every fixed $n \geq 3$. This follows from the NP-hardness proof of binary quadratic equations over the integers of [MA78]. This NP-hardness proof requires that det $A$ has a large square factor. However, CEP is polynomial time for isotropic, ternary forms with odd, squarefree determinant. For isotropic forms $A$ an isotropic vector $y \neq 0$, such that $y^t A y = 0$ can be found in polynomial time given the factorization of det $A$ [Si05]. Given an isotropic vector $y$ the equation $x^t A x = c$ can be solved in polynomial time for every $c \in \mathbb{Z}$ if det $A$ is odd and squarefree.

Proof of knowledge. Prover $P$ proves to verifier $V$ knowledge of $S$ such that $S^t A_1 S = A_0$ by iterating

1. $P$ computes and sends an LLL-reduced form $A' := T^t A_0 T$ for a randomized $T \in \text{GL}_n(\mathbb{Z})$, see [HS07].
2. $V$ sends a random one-bit challenge $b \in \{0, 1\}$.
3. $P$ sends the reply $R_b := S^t T \in \text{GL}_n(\mathbb{Z})$, and $V$ checks that $R_b^t A_1 R_b = A'$.
If a fraudulent $\bar{P}$ succeeds with $\bar{A}'$ and replies $\bar{R}_b$ for both $\bar{R}_0, \bar{R}_1$ he gets an equivalent private key $S^t := \bar{R}_1\bar{R}_0^{-1}$ satisfying $S^tA_1S^t = A_0$. This protocol is statistical zero-knowledge under reasonable heuristics.

Another proof of knowledge uses long challenges and can be transformed into an efficient public key signature scheme by replacing $V$ through a cryptographic hash function. This proof represents some $c \in \mathbb{Z}$ as $x^tA_bx = c = y^tA'y$. Here $x, y$ must and can be chosen such that the problem to extend $x, y$ to an equivalence transform $T \in \text{GL}_n(\mathbb{Z})$ still requires exponential time by known algorithms. In fact the reconstruction of $T$ requires to represent the determinant of some $(n-1)$-dimensional subform of $A'$ by the $(n-1)$-dimensional, adjoint form $A^R_b$ of $A_b$. No subexponential algorithm is known for this latter problem for any fixed $n \geq 4$. Most instances of this problem are subexponential for $n = 3$.

References


Low-end uniform hardness versus randomness tradeoffs for Arthur-Merlin games

Ronen Shaltiel

(joint work with Chris Umans)

In 1998, Impagliazzo and Wigderson [IW98] proved a hardness vs. randomness tradeoff for BPP in the uniform setting, which was subsequently extended to give optimal tradeoffs for the full range of possible hardness assumptions by Trevisan and Vadhan [TV02] (in a slightly weaker setting). In 2003, Gut Freund, Shaltiel and Ta-Shma [GSTS03] proved a uniform hardness vs. randomness tradeoff for AM, but that result only worked on the “high-end” of possible hardness assumptions.

In this work, we give uniform hardness vs. randomness tradeoffs for AM that are near-optimal for the full range of possible hardness assumptions. Following [GSTS03], we do this by constructing a hitting-set-generator (HSG) for AM with “resilient reconstruction.” Our construction is a recursive variant of the Miltersen-Vinodchandran HSG [MV99], the only known HSG construction with
this required property. The main new idea is to have the reconstruction procedure operate implicitly and locally on superpolynomially large objects, using tools from PCPs (low-degree testing, self-correction) together with a novel use of extractors that are built from Reed-Muller codes [SU01] for a sort of locally-computable error-reduction.

As a consequence we obtain gap theorems for AM (and AM ∩ coAM) that state, roughly, that either AM (or AM ∩ coAM) protocols running in time $t(n)$ can simulate all of EXP ("Arthur-Merlin games are powerful"), or else all of AM (or AM ∩ coAM) can be simulated in nondeterministic time $s(n)$ ("Arthur-Merlin games can be derandomized"), for a near-optimal relationship between $t(n)$ and $s(n)$. As in [GSTS03], the case of AM ∩ coAM yields a particularly clean theorem that is of special interest due to the wide array of cryptographic and other problems that lie in this class.

References


Towards Universal Semantic Communication

Madhu Sudan

(joint work with Brendan Juba)

Consider the following fantastic scenario: Earth has just started receiving some signals from outer space. These signals don’t seem like usual cosmic noise. Potentially an intelligent alien civilization is trying to make contact. How should Earth respond? How can we (earthlings) tell if the aliens are receiving our response and reacting to it? Are they really intelligent, or are we talking to sunspots? If they are intelligent, will we ever be able to achieve meaningful interaction in this setting?

Can these questions be tackled mathematically? The classical theory of communication, while founded solidly in mathematics, typically ignores the issue of
semantics of communication, and has focused principally on quantitative measures in syntactic settings. Increasingly, however, it is becoming clear that practical challenges to communication arise due to semantic gaps between senders and receivers. The fictional problem above, merely, carries this gap to the extreme.

In this work, we attempt to describe how the theory of computational complexity can shed light on such interactions. The principal goal is to figure out how some of the nebulous notions, such as intelligence and understanding, should be defined in this setting. We assert that in order to communicate “successfully”, the communicating players should be explicit about their goals - what the communication should achieve. We show that when the goals are explicit the communicating players can achieve meaningful interaction, provided the players are capable of satisfying the goals, and cooperative, under reasonable, mathematical, definitions of these notions.

References


Integrality Gaps for Vertex Cover in Lovasz-Schrijver Hierarchies

LUCA TREVISAN

(joint work with Grant Schoenebeck, Madhur Tulsiani)

We study linear and semidefinite programming relaxations of Vertex Cover arising from repeated applications of the “lift-and-project” method of Lovasz and Schrijver [5] starting from the standard linear programming relaxation.

For linear programs (LS), Arora, Bollobas, Lovasz and Tourlakis [1] prove that the integrality gap remains at least $2 - \epsilon$ after $\Omega_1(\log n)$ rounds, where $n$ is the number of vertices, and Tourlakis [6] proves that integrality gap remains at least $1.5 - \epsilon$ after $\Omega_1((\log n)^2)$ rounds. We prove that the integrality gap remains at least $2 - \epsilon$ after $\Omega_1(n)$ rounds.

For semidefinite programs (LS+), Goemans and Kleinberg [4] prove that after one round the integrality gap remains arbitrarily close to 2. Charikar [2] proves an integrality gap of 2 for a stronger relaxation that is, however, incomparable with two rounds of LS+ and is strictly weaker than the relaxation resulting from a constant number of rounds. Georgiou et al. [3] show that the integrality gap remains $2 - o(1)$ after $\Omega(\sqrt{\log n/log\log n})$ rounds. We prove that the integrality gap remains at least $7/6 - \epsilon$ after $\Omega_1(n)$ rounds.

References

Complexity Theory

Evolution

Leslie G. Valiant

We suggest a quantitative model of evolution for the purpose of studying how representations of complex functions can evolve from simpler ones within realistic population sizes and numbers of generations [1]. Evolution is treated as a form of computational learning, in which the course of learning depends only on the fitness of the hypothesis on the aggregate of the examples, and not otherwise on the examples. We formulate a notion of evolvability for different classes of functions. It is shown that in any one phase of evolution monotone Boolean conjunctions and disjunctions are evolvable over the uniform distribution, while Boolean parity functions are not. The framework also suggests how a wider range of issues in evolution might be quantified. We also suggest that the process of biological evolution over multiple phases should be viewed as evolvable target pursuit, which consists of a series of evolutionary phases, each one pursuing a target that is evolvable in our technical sense, each target being rendered evolvable by the serendipitous combination of the environment and the outcome of previous evolutionary phases.

REFERENCES


One-way multi-party communication lower bound for pointer jumping with applications

Emanuele Viola

(joint work with Avi Wigderson)

In this paper we study the one-way multi-party communication model, in which every party speaks exactly once in its turn. For every fixed k, we prove a tight lower bound of \( \Omega \left( n^{1/(k-1)} \right) \) on the probabilistic communication complexity of pointer jumping in a k-layered tree, where the pointers of the \( i \)-th layer reside on the forehead of the \( i \)-th party to speak. The lower bound remains nontrivial even
for $k = (\log n)^{1/3}$ parties. Previous to our work a lower bound was known only for $k = 3$, and in very restricted models for $k > 3$. Our results have the following consequences to other models and problems, extending previous work in several directions.

The one-way model is strong enough to capture general (non one-way) multi-party protocols of bounded rounds. Thus we generalize to this multi-party model results on two directions studied in the classical 2-party model (e.g [PS, NW]). The first is a round hierarchy: We give an exponential separation between the power of $r$ and $2r$ rounds in general probabilistic $k$-party protocols, for any fixed $k$ and $r$. The second is the relative power of determinism and nondeterminism: We prove an exponential separation between nondeterministic and deterministic communication complexity for general $k$-party protocols with $r$ rounds, for any fixed $k, r$.

The pointer jumping function is weak enough to be a special case of the well-studied disjointness function. Thus we obtain a lower bound of $\Omega (n^{1/(k-1)})$ on the probabilistic complexity of $k$-set disjointness in the one-way model, extending a similar lower bound for the weaker simultaneous model, in which parties simultaneously send one message to a referee [BPSW].

Finally, we infer an exponential separation between the power of different orders in which parties send messages in the one-way model, for every fixed $k$. Previous to our work such a separation was only known for $k = 3$ [NW].

Our lower bound technique, which handles functions of high discrepancy, may be of independent interest. It provides a “party-elimination” induction, based on a restricted form of a direct-product result, specific to the pointer jumping function.

This work will appear in FOCS 2007.

References


Nechiporuk Bounds for the Middle Bit of Multiplication

INGO WEGENER

(joint work with Philipp Woelfel)

Other results on restricted branching programs for $MM_N$, the middle bit of multiplication, have revealed a lot about the subfunction structure of this function. This leads to the aim to investigate the best possible bounds on the branching program size and the formula size of $MM_N$ obtainable by Nechiporuk's lower bound technique. We prove bounds of size $\Omega(n^{3/2} / \log n)$ and $\Omega(n^{3/2})$ respectively and prove
that these bounds can be improved by not more than an $n^{1/6}$-factor. The results have been presented at the conference Computational Complexity (2005) and the full version is accepted for publication in the journal Computational Complexity.

**Network Extractor Protocols and Three-Source Extractors**

**David Zuckerman**

(joint work with Xin Li, Anup Rao)

We design several efficient one-round network extractor protocols, which extract private randomness over a network with faulty players when each player has a single, weak random source of sufficient min-entropy. As a corollary, we derive efficient protocols for Byzantine agreement and leader election (and hence the equivalent collective coin-flipping) in the full information model. Our robust protocols run in just one more round than the corresponding protocols with perfect randomness. Our results significantly improve those of Goldwasser, Sudan, and Vaikuntanathan [1].

In a synchronous network, if each of $p$ players has a weak source with min-entropy rate greater than $1/2$, then we essentially match the bounds for perfect randomness: Byzantine agreement tolerating a $1/3 - \alpha$ fraction faulty players in $O(\log p)$ rounds, and leader election tolerating a $1/2 - \alpha$ fraction faulty players in $\log^* p + O(1)$ rounds, for any constant $\alpha > 0$. In a synchronous network, if each player’s $n$-bit source of randomness has $n^{\Omega(1)}$ min-entropy, then the bounds drop to $1/4 - \alpha$ and $1/3 - \alpha$, respectively. In an asynchronous network, if each player has access to a source with polynomial min-entropy (though $1/3$ of the players need shorter sources than the others), then our Byzantine agreement protocol tolerates a $1/18 - \alpha$ fraction of faulty players.

Extractors for independent sources are crucial to our results. In particular, our results for asynchronous protocols rely on a new extractor for three independent sources. Two of the sources must have $n$ bits with min-entropy at least $n^\gamma$; the third must have $n^{\gamma/c}$ bits with min-entropy at least $\log^{10} n$. (Here $c$ is an absolute constant and $\gamma > 0$ is arbitrary.) Previously, extractors for independent sources with min-entropy $n^\gamma$ required $O(1/\gamma)$ sources [2].

**References**
