A Concise Research Summary

Iddo Tzameret

January 2017

My main area of research is the foundations of computer science, computational complexity, satisfiability (in practice and theory), and applications of logic in computer science. I am interested in any kind of research in the foundations of computing, both in the ***conceptual*** aspect of the field, namely, the modeling of natural computational phenomena, or the modeling of natural phenomena through a computational lens; its ***applications*** and interactions with different areas of study; as well as in its more ***mathematical*** aspect, namely, advancing our understanding of the fundamental limits of efficient computation, by establishing lower bounds on various models of computations.

1 Short Summary

In recent years I have been conducting research on the theory of computation, with an emphasis on computational complexity and proof complexity. My main contributions are in applying methods from computational complexity, algebraic complexity and logic in the area of efficient reasoning and proof complexity. In that respect, I have been developing the algebraic proofs regime [10, 20, 11, 18, 17, 12, 14, 16, 9], and propositional proofs under approximations and average-case proof complexity [7, 15, 21]. My research was funded by the Natural National Science Foundation of China (NSF China).

1.1 Background

The study of the limits of efficient computation lies at the frontier of contemporary science. The significance of this problem stems both from its clear fundamental aspect as a question about the nature of the physical world, the nature of mathematics and their interplay, as well as from its provable practical relevance to the contemporary information era, in which billions of explicit computational tasks are being conducted every minute.

When building a *theoretical foundation* for efficient computation, we are necessarily drawn to make ideal simplifications regarding what is computation to start with, and what kind of computations are considered efficient. Fortunately, the foundational groundwork addressing many of these questions has already been conducted in the 20th century, through the works of mathematicians, logicians, and computer scientists, such as Kurt Gödel and Alan Turing. Of a special relevance to my research is the extremely influential work of Stephen A. Cook (and other prominent scientists such as Leonid Levin) who has laid the foundations for what we consider nowadays to be the *theory of efficient computation*, and the still somewhat enigmatic phenomenon of NP-completeness. Problems we take as efficiently computable (at least in an idealized manner) are problems solvable by a deterministic algorithm that runs in polynomial-time, where the polynomial is in the size of the input to the algorithm.

An NP problem is a problem, or a set A of strings (i.e., a *language*) we wish to determine, whose instances have short certificates (or equivalently, *witnesses* or *proofs*). In other words, given a string x of

length s, x is in A if and only if there exists a certificate y of length polynomial in s that can be checked efficiently for correctness. Now, the question whether problems in NP, namely those sets whose instances have short and efficiently verifiable certificates, also have fast algorithms is one of the central problems in computer science, and contemporary science by large. To put it differently, the question is to determine whether verifying a (correct) solution is algorithmically equivalent to *finding* a correct solution (i.e., finding the certificate). Clearly, this question, or in its formal name *the P versus NP problem*, is of great practical and scientific value.

This problem is known to be extremely difficult by itself, and it has obtained both intensive popular interest and scientific attention, while giving birth to many new ideas, and scientific branches within the theory of efficient computation. One such direction, which was initiated by the fundamental work of S. Cook [4] is that of ***Proof Complexity***. This direction is very simple to explain in the context of computational complexity described above: in proof complexity we take problems, or languages A, that we assume do *not* have short certificates, and try *formally to prove* that if we *restrict* the way the certificates are being written, then indeed there are no short certificates for A; that is, there will be instances $x \in A$ that require very large certificates y (when y is written in the restricted manner we chose).

Let me give an example. Assume we get m integer inequalities

$$\langle A_1, \mathbf{x} \rangle \leq b_1, \ldots, \langle A_m, \mathbf{x} \rangle \leq b_m,$$

where $\langle \cdot, \cdot \rangle$ is the inner product, the A_i's are integers vectors and the b_i's are integer numbers and $\mathbf{x}$ are the indeterminates. Are there short certificates that there is an *integer* solution to this system of inequalities? It is known that the answer is positive: if the system of inequalities is solvable then the solution itself, namely the integer-assignment to the indeterminates $\mathbf{x}$ will serve as the short certificate (it can be shown that this certificate is indeed small in the size of the input system).

But what about the following problem: **are there short certificates that the system of inequalities is *unsatisfiable*?** It is believed that there are *no* such short certificates of unsatisfiability. However, to prove this is apparently extremely difficult, and so no real proofs of this fact is known at this stage. Indeed, we know that determining that such systems of integer inequalities have no solutions is a coNP-complete problem (and thus showing it does not have short certificates would mean separating NP from coNP, and thus separating also P from NP).

In Proof Complexity what we usually do is to ***restrict*** the way that certificates for a given problem (such as the problem considered above) are written. By putting such restrictions on the way the certificates are written we achieve two things: first, we have a better chance in actually understanding the complexity of such certificates, namely, we are able to understand sometimes which instances are easy (have short certificates) and which instances are *hard* (require large certificates). And second, we can show that instances requiring large certificates (if we restrict the way we write the certificates) are *hard for certain algorithms*; that is, hard for those algorithms whose run induce a (restricted) certificate for the instance.

The most prominent example for the second point above, is the one between lower bounds on *resolution refutations* of unsatisfiable CNF formulas and lower bounds on the run-time of practical (DPLL-based) contemporary SAT-solvers [1] (i.e., tools for deciding, given an input CNF formula, whether the input is satisfiable or not).

The subject of Proof Complexity is thus a fascinating subject and one of the most fundamental natural within the theory of computing, which is at the heart of computational complexity theory. It can be viewed as part of Concrete Complexity, that is, a part of the directions that seek to provide unconditional limitations on concrete computational models (together with Boolean and Algebraic Circuit Complexity and Communication Complexity).

1.2 Overview of research themes

I have a broad interest in all areas pertaining to the theory of computation; and I have a special interest in questions in the area of Proof Complexity. My research revolves around finding new directions towards the lower bounds questions in computational complexity, proof complexity and connecting these questions to algorithmic questions, algebraic complexity, pure algebra and logic.

For most part my work so far revolves around three main inter-related themes:

(i) ***Polynomial identities***: Studying the complexity of nondeterministic algorithms for identifying polynomial identities; and specifically, proofs establishing *polynomial identities* (in contrast to propositional tautologies), with connections to derandomization of probabilistic algorithms and its applications to the area of feasible reasoning. Complexity of polynomial identities over matrices and matrix algebras.

(ii) ***Propositional proof complexity***: The study of proof systems establishing propositional *tautologies* by means of manipulating polynomials over a field or Boolean circuits. This is part of the main thread in contemporary Proof Complexity.

(iii) ***Approximate, probabilistic and average-case reasoning; Random k-SAT refutation algorithm***: Developing frameworks and concrete computational models for approximate, probabilistic and average-case reasoning and studying their efficiency and connections with computational complexity, *proof-search algorithms* and refutations algorithms for random formulas.

1.3 Summery of recent research projects

Together with Hrubeš, I have developed the theory of *arithmetic proofs of polynomial identities* [10, 11] in relation to the Polynomial Identity Testing problem from algebraic complexity and derandomization theory. This led us [11] to the resolution of a long-standing open problem, first posed by S. Cook, (cf. [3, 19]), in the area of efficient reasoning; namely, establishing short propositional proofs for the basic determinant identities (and loosely speaking, for all statements of linear algebra).

In a joint project with Stephen Cook [22], we worked on a strengthening of the results from [11].

Together with my student Fu Li, we investigated in [13] the complexity of generating matrix identities with the hope to develop a novel algebraic technique for establishing lower bounds on strong proof systems (one of the fundamental open problems in complexity). We show that for any d and any finite set of generators of the identities of $d \times d$ matrix algebras (over a field), there is a family of polynomials that requires many (namely, $\Omega(n^{2d})$) generators to generate. Under further assumptions, this may lead to very strong (up-to exponential) lower bounds on certain proofs of polynomial identities, and hence significantly advance our understanding of the limits of efficient provability and computation.

Another important direction I have been involved in is the study of *average-case* unsatisfiability. Together with Müller [15], I have established short propositional refutations for random 3CNF formulas in a very weak propositional proof system. This places a stream of recent results on efficient refutation algorithms using *spectral arguments*—beginning in the work of Goerdt and Krivelevich (2001) and culminating with the important results of Feige, Kim and Ofek (2006)—within the framework of propositional proofs. Loosely speaking, we show that all these refutation algorithms and witnesses, considered from the perspective of propositional proofs, can be simulated efficiently within a very restricted proof systems.

In my work [21], I further show that tackling these spectral-based refutation algorithms from the perspective of propositional-proofs may have applications in actually solving an important open problem in the frontiers of refutation algorithms: improving the current best efficient *deterministic* refutation algorithm for

random 3SAT (the latter works only for the clause-to-variable density $\Omega(n^{0.5})$; see [8]) so that it will work already for $\Omega(n^{0.4})$ clause-to-variable density.

References

[1] *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, 2009. 1.1

[2] Eric Allender, George Davie, Luke Friedman, Sam Hopkins, and Iddo Tzameret. Kolmogorov complexity, circuits, and the strength of formal theories of arithmetic. *Chicago Journal of Theoretical Computer Science*, (5):1–15, 2013.

[3] Maria Luisa Bonet, Samuel R. Buss, and Toniann Pitassi. Are there hard examples for Frege systems? In *Feasible mathematics, II (Ithaca, NY, 1992)*, volume 13 of *Progr. Comput. Sci. Appl. Logic*, pages 30–56. Birkhäuser Boston, Boston, MA, 1995. 1.3

[4] Stephen A. Cook. The complexity of theorem proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, pages 151–158. ACM, New York, 1971. 1.1

[5] Nachum Dershowitz and Iddo Tzameret. Gap embedding for well-quasi-orderings. *Electr. Notes Theor. Comput. Sci.*, 84:80–90, 2003.

[6] Nachum Dershowitz and Iddo Tzameret. Gap embedding for well-quasi-orderings. In Elaine Pimentel Ruy de Queiroz and Lucilia Figueiredo, editors, *Proceedings of the 10th Workshop on Logic, Language, Information and Computation (Wollic'03)*, volume 84 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.

[7] Nachum Dershowitz and Iddo Tzameret. Complexity of propositional proofs under a promise. *ACM Transactions on Computational Logic*, 11(3):1–30, 2010. Preliminary version appeared in *ICALP* '07, pp. 291–302. 1

[8] Uriel Feige and Eran Ofek. Easily refutable subformulas of large random 3CNF formulas. *Theory of Computing*, 3(1):25–43, 2007. 1.3

[9] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 32:1–32:17, 2016. 1

[10] Pavel Hrubeš and Iddo Tzameret. The proof complexity of polynomial identities. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 41–51, 2009. 1, 1.3

[11] Pavel Hrubeš and Iddo Tzameret. Short proofs for the determinant identities. *SIAM J. Comput.*, 44(2):340–383, 2015. (A preliminary version appeared in Proceedings of the 44th Annual ACM Symposium on the Theory of Computing (STOC'12)). 1, 1.3

[12] Fu Li and Iddo Tzameret. Generating matrix identities and proof complexity lower bounds. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:185, 2013. 1

[13] Fu Li and Iddo Tzameret. Witnessing matrix identities and proof complexity. *International Journal of Algebra and Computation*, 28(2):217–256, 2018. 1.3

[14] Fu Li, Iddo Tzameret, and Zhengyu Wang. Non-commutative formulas and Frege lower bounds: a new characterization of propositional proofs. In *Proceedings of the 30th Computational Complexity Conference (CCC), June 17-19, 2015*, 2015. 1

[15] Sebastian Müller and Iddo Tzameret. Short propositional refutations for dense random 3CNF formulas. *Annals of Pure and Applied Logic*, 165:1864–1918, 2014. Extended abstract in Proceedings of the 27th Annual ACM-IEEE Symposium on Logic In Computer Science (LICS), 2012. 1, 1.3

[16] Tonnian Pitassi and Iddo Tzameret. Algebraic proof complexity: Progress, frontiers and challenges. *ACM SIGLOG News*, 3(3), 2016. 1

[17] Ran Raz and Iddo Tzameret. Resolution over linear equations and multilinear proofs. *Ann. Pure Appl. Logic*, 155(3):194–224, 2008. 1

[18] Ran Raz and Iddo Tzameret. The strength of multilinear proofs. *Computational Complexity*, 17(3):407–457, 2008. 1

[19] Michael Soltys and Stephen Cook. The proof complexity of linear algebra. *Ann. Pure Appl. Logic*, 130(1-3):277–323, 2004. 1.3

[20] Iddo Tzameret. Algebraic proofs over noncommutative formulas. *Inf. Comput.*, 209(10):1269–1292, 2011. 1

[21] Iddo Tzameret. Sparser random 3-SAT refutation algorithms and the interpolation problem. In *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP), track A*, 8-11 July 2014. 1, 1.3

[22] Iddo Tzameret and Stephen A. Cook. Uniform, integral and efficient proofs for the determinant identities. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12, 2017. 1.3