Research Statement

Anamay Tengse*
anamay.tengse@gmail.com

Introduction

I am mainly interested in complexity theory, a field that deals with determining the cost of computational tasks, in terms of various resources like time, space, randomness, etc. My research thus far has been in *algebraic complexity theory*, where the focus is on tasks that are naturally modeled as polynomials, like matrix multiplication, determinant, and so on.

For any fixed polynomial, perhaps the most natural way to quantify the cost of computing it is to ask, *"how many additions and multiplications are required to compute its value on* any *input?"*. The computational model of *algebraic circuits* captures precisely this intuition. An algebraic circuit starts with the variables and constants (from some field), and builds a polynomial *syntactically* using additions and multiplications. The cost of computing any fixed polynomial, is then captured by the *smallest circuit* — one that uses fewest operations — computing it.

In algebraic circuit complexity, we usually deal with polynomials whose degree depends *polynomially* [1] on the number of variables, which are sometimes called *low-degree polynomials*. Therefore the complexity of a polynomial, size of the smallest circuit computing it, is generally expressed solely in terms of the number of variables it depends on. The "easy polynomials" are therefore n-variate (low-degree) polynomials that have circuits of size $\text{poly}(n)$ computing them. It can be shown that a random polynomial — each of whose coefficients is picked uniformly at random — is a *hard polynomial* with high probability. This brings us to the following central question in algebraic circuit complexity, that of finding *explicit* hard polynomials.

> **Lower Bounds**: *Find an explicit n-variate polynomial (for every $n \in \mathbb{N}$) that requires circuits of size $n^{\omega(1)}$ to compute it.*

Valiant, in his foundational work [Val79], formalized the notions of *efficiently computable polynomials* and *efficiently definable polynomials*, which are now denoted by the complexity classes VP and VNP, respectively. The class VP exactly matches our above definition of "easy polynomials", and the class VNP turns out to be a good analogue for the class of "explicit polynomials", thus translating the *lower bounds* question to a succinct one: *"Is* $\mathsf{VP} = \mathsf{VNP}$*"*?

*Tata Institute of Fundamental Research, Mumbai, India

[1] The notion of asymptotic behavior can be formalized via the language of *polynomial families*. We will avoid doing that for the ease of exposition.

Let us now shift our perspective, and ask *"given a circuit, what can be said about the polynomial it computes?"*. Perhaps the simplest task of this kind is to determine whether or not the given circuit computes just the zero polynomial; this is called the *identity testing* question.

It is a well-known fact that any nonzero n-variate degree d polynomial evaluates to a nonzero value at a random point from a large enough "grid", e.g. $[2d]^n$, with high probability [Ore22, DL78, Zip79, Sch80]. This fact immediately gives an efficient randomized algorithm for the identity testing question, even for those low-degree polynomials that may not have small circuits. Therefore it is reasonable to expect a "derandomization" of this algorithm at least for VP, with a *deterministic* identity test that runs in time polynomial in the size of the input circuit. Additionally, the above randomized algorithm is also a *blackbox polynomial identity test (blackbox PIT)*, that is it only evaluates the given circuit at some points, as opposed to a *whitebox PIT*, which can also access the underlying graph of the circuit. Therefore, an interesting question is whether VP has an *efficient, deterministic, blackbox PIT*.

As it turns out, if we are designing a *deterministic blackbox PIT*, then we can do away with the power of adaptive querying, thus reducing the task to that of finding the corresponding set of evaluation points, a *hitting set*. Formally, a set of evaluation points $\mathcal{H}$ is said to be *hitting set* for a class of polynomials $\mathcal{C}$, if for every nonzero $f \in \mathcal{C}$ there is some point $h \in \mathcal{H}$ such that $f(h) \neq 0$. Just as in the case of hard polynomials, a random set of $\text{poly}(s)$ many evaluation points is a hitting set for circuits of size s [HS80, For14]. The challenge is therefore to find an *explicit* hitting set of small size, as stated below.

> **Hitting Sets**: *Find an explicit hitting set of size* $\text{poly}(n, d, s)$ *for the class of n-variate, degree d polynomials that are computed by circuits of size s.*

Current Status The previously mentioned work of Valiant [Val79] is widely regarded as the beginning of algebraic circuit complexity, and there has been a fair amount of progress in the area since then. It is therefore surprising that the progress on both the questions mentioned above has been marginal. We do not know of any explicit polynomial that requires circuits of size $n^{1.01}$; the best known lower bound is $\Omega(n \log n)$ [Str73, BS83]. At the same time, the best known *explicit* hitting sets for circuits have *exponential* size, which are just large enough grids like $[2d]^n$, that use no information about the complexity of the polynomials. It is therefore natural to first address these questions in some restricted settings, and then try to infer a strategy to attack the general questions.

Contributions

My work so far has focused on understanding some of the interesting restricted settings, mostly in the context of the hitting set question, and furthermore in investigating the challenges in solving the general questions. In particular, I have been part of the following contributions to algebraic circuit complexity.

- **Extending the scope of known constructions**: One way to make progress on the identity testing front is to show that the explicit hitting sets given by known techniques in fact give hitting sets for a larger class. In a joint work with Saptharishi [ST18], we show that the known hitting set constructions [AGKS15, GKST15, GKS16] for the class of *non-commutative ABPs*, in fact extend to the class of *unique parse tree (UPT) circuits* [LMP16], a model known to be strictly more powerful than ABPs.

- **Near-optimal hitting sets from minuscule improvements**: As mentioned before, the best known hitting sets for n-variate, degree d, size s circuits have exponential size, $d^{O(n)}$. A natural question here is to ask what happens if we are able to make a minuscule improvement in the exponent, say hitting sets of size $s^{1.99}$ for *bivariate*, size s, degree s circuits (or formulas), for all large s? In a joint work with Kumar and Saptharishi [KST19], we show that this will lead to nearly $\text{poly}(s)$ sized hitting sets for size s circuits (or formulas, respectively), for all large s. We achieve this by building on the framework of *bootstrapping of hitting sets* introduced by Agrawal, Ghosh and Saxena [AGS18], where they had obtained the same conclusion starting with hitting sets of size at most $s^{k^{0.499}}$ for k-variate, size s circuits, for any large enough constant k and all large s.

- **Existence of *algebraically natural proofs***: The lack of success in questions for algebraic circuits despite significant progress in several restricted settings has lead to research that investigates the (in)ability of the current proof strategies to prove strong circuit lower bounds. Analogous to the work of Razborov and Rudich [RR97] for *boolean circuits*, Forbes, Shpilka and Volk [FSV18], and Grochow, Kumar, Saks and Saraf [GKSS17] proposed the framework of *algebraically natural proofs*, which they showed covers almost all the known strategies for proving algebraic circuit lower bounds. Using this framework, they showed that if VP — the class of all *poly-sized* circuits — has *succinct hitting sets*, then there are no algebraically natural proofs against VP. Forbes *et al.* [FSV18] also provided some evidence against the existence of natural proofs, by constructing *succinct hitting sets* for several restricted classes.

  However, the question of whether VP indeed has natural proofs still remains open. In a joint work with Chatterjee, Kumar, Ramya and Saptharishi [CKR+20], we provide some evidence *for* the existence of algebraically natural proofs for VP. In particular, we show that when we restrict ourselves to polynomials that have "moderately large" integer coefficients, there exist *efficiently computable* polynomial equations, that can distinguish such polynomials in VP from many others outside it.

I will now elaborate on these results briefly, before moving on to describe some of the questions that I would like to explore in the near future.

Hitting Sets for Non-commutative Models

Non-commutative models compute polynomials over variables that do not commute under multiplication (e.g. $xy \neq yx$); monomials are therefore *words* on the alphabet of the variables. A

particular use-case could be polynomials that operate on matrices, instead of field scalars. Non-commutativity restricts the number of different ways in which a monomial can be expressed, and therefore one expects that it should be relatively easier to obtain lower bounds or PITs for such models. This indeed turns out to be true.

The seminal work of Nisan [Nis91] that initiated the study of non-commutative models, gives *exponential* lower bounds against non-commutative *algebraic branching programs (ABPs)*, a weaker variant of circuits. Nisan showed that the ABP-complexity of any non-commutative polynomial is *exactly* characterized by the rank of its *coefficient matrices*, an idea very similar to that of a *communication matrix* from communication complexity. Building on the Nisan's characterization, efficient whitebox PITs and *quasipolynomial* ($n^{O(\log(n))}$) sized hitting sets were obtained for non-commutative ABPs [RS05, FS13]. An analogous *commutative* model is that of Read-once Oblivious ABPs (ROABPs) [FS13], for which similar results have been obtained [AGKS15, GKST15, GKS16].

Recently, Lagarde, Malod and Perifel [LMP16] extended Nisan's work and obtained an exact characterization for a model called *Unique Parse Tree (UPT) circuits*, which they showed lies strictly between ABPs and circuits in terms of computational power. Their work, combined with a follow-up work by Lagarde, Limaye and Srinivasan [LLS17], extended the then known lower bounds and whitebox PITs for ABPs [Nis91, RS05, GKST15] to UPT circuits. However, no non-trivial hitting set constructions were known for UPT circuits and the corresponding related models.

In a joint work with Saptharishi [ST18], we gave a quasipolynomial time blackbox PIT for UPT circuits, building on the results of Agrawal, Gurjar, Korwar and Saxena [AGKS15]. We also extended the other then known hitting set constructions [GKST15, GKS16], for similar counterparts of UPT circuits and their commutative analogs.

Bootstrapping Hitting Sets

My next work studied the problem of constructing explicit hitting sets for general algebraic circuits. Apart from its connections to the hardness question, the task of obtaining small explicit hitting sets for easily computable polynomials is interesting in its own right as a derandomization problem owing to the crucial use of PITs in well-known results (e.g. [MVV87, LFKN90, AKS04]). It is then slightly surprising that the best known explicit hitting sets for n-variate, degree-d polynomials computable by circuits of size s are as large as $d^{O(n)}$, a trivial consequence of the randomized algorithm, as mentioned before. As a key step towards understanding hitting sets for algebraic circuits, Agrawal, Ghosh and Saxena [AGS18] observed a surprising phenomenon about hitting sets that they called *bootstrapping*.

They essentially showed that if we make a "marginal" improvement in explicit hitting sets for constant variate circuits, then it would imply hitting sets for algebraic circuits, that are *almost polynomial* in size. More formally, they proved that explicit hitting sets of size as large as $s^{k^{0.499}}$, for k-variate circuits of size and degree s, for some large constant k and all large s, would imply explicit hitting sets of size $s^{\text{tiny}(s)}$ for s-variate circuits of size and degree s, for all large s, and a

very slow growing function $\text{tiny}(\cdot)$[2].

Although their result asked for a fairly mild improvement in the trivial hitting set of size $s^{O(k)}$, some natural questions that come up are as follows.

1. Is it possible to replace $k^{0.499}$ by something even closer to k, like $k^{0.999}$ or $k-1$?
2. Can we start the *bootstrapping* procedure from a much smaller constant k?
3. Is a similar result true for subclasses of circuits, like formulas or ABPs?

In a joint work with Kumar and Saptharishi [KST19], we built on the techniques of Agrawal *et al.* [AGS18] to answer all these questions in the affirmative while maintaining the same conclusion and obtained a near-optimal version of their result. We showed that explicit hitting sets of size as large as $s^{k-0.01}$, for k-variate circuits of size and degree s, for any $k \geq 2$ and all large s, would imply explicit hitting sets of size $s^{\text{tiny}(s)}$ for s-variate circuits of size and degree s, for all large s. Furthermore, our statement also holds for subclasses of circuits, i.e. algebraic branching programs (ABPs) and algebraic formulas, and also works over fields of all characteristics. Both these things were not known to be true for the result of Agrawal *et al.* [AGS18].

Proving Lower Bounds

As mentioned before, even the question of proving super-linear lower bounds against circuits has remained open till now. On the other hand, we have seen several notable achievements in various restricted models [Yeh19, Sri19, Raz06, Raz10, Nis91, LMP16, GKKS14]. As a result, there have been attempts at getting a better sense of the ability of the current approaches to prove lower bounds for algebraic circuits in general, in order to design new approaches that might work against the general models.

A crucial work of this kind in *boolean circuit complexity* is that of Razborov and Rudich [RR97] where they introduced the notion of *natural proofs*. They showed that most of the then known approaches to proving boolean circuit lower bounds followed a certain template, which they called the *natural proofs* framework. They then went on to show that if sufficiently hard *one way functions* exist, then there were no natural proofs against P/poly, the class of polynomial sized boolean circuits. Since the existence of such one way functions is widely believed in the community, this essentially meant that any proof of a super-polynomial boolean circuit lower bound would have to deviate from the *natural proofs* framework.

Recently, the works of Grochow, Kumar, Saks and Saraf [GKSS17], and Forbes, Shpilka and Volk [FSV18] adapted the ideas of Razborov and Rudich to the algebraic world, via the notion of *algebraically natural proofs*. They showed that most of the current proofs of algebraic circuit lower bounds are in fact *algebraically natural proofs*. Additionally, they showed that if VP — the class of polynomial sized algebraic circuits — had *succinct hitting sets*, then any proof of a super-polynomial lower bound must avoid such a template of *algebraically natural proofs*. As evidence

[2] $\text{tiny}(s) = \exp(\exp(O(\log^* s)))$

against the existence of algebraically natural proofs, Forbes *et al.* [FSV18] showed that several known hitting set constructions for restricted models could in fact be made *succinct*, even with restricted models.

Interestingly, explicit hitting sets are known to imply (almost) explicit lower bounds [HS80, Agr05, KI04], which then makes the fact that *succinct* hitting sets imply a potential barrier towards proving lower bounds, a bit counter-intuitive. It therefore makes sense to investigate whether the positive connections ([HS80, Agr05, KI04]) help us obtain any interesting lower bound proofs.

In a joint work with Chatterjee, Kumar, Ramya and Saptharishi [CKR+20], we addressed this question and gave evidence *for* the existence of *algebraically natural proofs*. An algebraically natural proof against a circuit class $\mathcal{C}$ is a non-zero polynomial P that outputs zero, whenever it is fed the list of coefficients of a polynomial from $\mathcal{C}$, thus "distinguishing" $\mathcal{C}$ from other polynomials. Furthermore, the polynomial itself needs to be computable efficiently. We showed the existence of such a polynomial P which distinguishes polynomials in VP that have small integer coefficients, from many other polynomials with small coefficients.

It should be noted that the restriction of small coefficients is only on the polynomial and *not on the circuits*. Thus for instance, if one shows that such a "distinguisher" polynomial P does not vanish on some polynomial in VNP (e.g. permanent), then VP $\neq$ VNP. Also, while we give a *non-constructive* proof of the existence of "efficiently computable distinguishers", the non-constructiveness comes solely from the use of *non-explicit* hitting sets. Thus, with pieces of evidence pointing in either directions, whether or not VP has algebraically natural proofs remains an interesting open question.

Future Directions

Here are some concrete open questions that are connected to my research till now, which I plan to investigate in the upcoming years.

- **Natural Proofs for** VP: An obvious question that comes out of the literature on algebraically natural proofs and related concepts [AD08, GKSS17, FSV18, CKR+20] is whether there are efficiently computable equations ("distinguishers") for the class VP. While finding an explicit equation will almost be as hard as proving super-polynomial lower bounds, can we expect a proof of existence like that in our recent work? Specifically for the purpose of proving lower bounds against VP, can we find some other subclass inside VP for which we can obtain *explicit* equations?

- **Constant Variate Circuits**: There are several results highlighting the connections between explicit lower bounds and hitting sets, for poly(n) sized n-variate circuits and poly(n) sized *constant* variate circuits [SS94, Koi11, KST19, GKSS19, And20]. At the same time, few candidate approaches are known for proving univariate circuit lower bounds. It is likely that some facts that are specifically known for univariate polynomials, e.g. Descartes rules of

signs, might help in these questions, and could also potentially lead to new techniques even for the multivariate setting.

- **Finer Separations within ROABPs**: The simplest model which has resisted efficient hitting sets, is perhaps that of *sums of powers of linear forms*, which was introduced by Saxena [Sax08]. Tight lower bounds are known for this model, for a monomial, while the best known hitting sets have size $n^{O(\log\log n)}$ [FSS14, GKS16]. From the same works, one can obtain a connection to *Read-once Oblivious ABPs (ROABPs)*, where obtaining $\text{poly}(s)$ sized hitting sets for size s ROABPs on $O(\log s)$ variables would give efficient hitting sets for sums of powers of linear forms. Moreover, the ROABPs that arise from the above connection are just *sums of products of univariates*. Surprisingly, while this model seems like a heavily restricted subclass of ROABPs, this is not known to be true. Apart from being an intriguing problem in its own right, I think such a separation could potentially solve the hitting set question for sums of powers of linear forms, or at least suggest some approaches for the same.

Branching Out Several ideas that I have come across in algebraic complexity theory are motivated from other areas of theoretical computer science, like pseudorandomness, boolean circuit complexity, communication complexity, etc. I am therefore excited about understanding these areas better, and exploring such connections, in the coming years.

Apart from this, my core interest has always been solving problems, with whatever tools and techniques that are available to me. I am therefore always looking forward to such opportunities, irrespective of the particular areas that the problems come from.

References

[AD08] Scott Aaranson and Andrew Drucker. Arithmetic natural proofs theory is sought. Shtetl Optimized: Scott Aaranson's Blog, 2008.

[AGKS15] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-Sets for ROABP and Sum of Set-Multilinear Circuits. *SIAM Journal of Computing*, 44(3):669–697, 2015. Pre-print available at `arXiv:1406.7535`.

[Agr05] Manindra Agrawal. Proving Lower Bounds Via Pseudo-random Generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105. Springer, 2005.

[AGS18] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, pages 1166–1179. ACM, 2018. `eccc:TR18-035`.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[And20] Robert Andrews. Algebraic Hardness versus Randomness in Low Characteristic. *CoRR*, abs/2005.10885, 2020. Pre-print available at arXiv:2005.10885.

[BS83] Walter Baur and Volker Strassen. The Complexity of Partial Derivatives. *Theoretical Computer Science*, 22:317–330, 1983.

[CKR+20] Prerona Chatterjee, Mrinal Kumar, C. Ramya, Ramprasad Saptharishi, and Anamay Tengse. On the Existence of Algebraically Natural Proofs. *CoRR*, abs/2004.14147, 2020. Pre-print available at arXiv:2004.14147.

[DL78] Richard A. DeMillo and Richard J. Lipton. A Probabilistic Remark on Algebraic Program Testing. *Information Processing Letters*, 7(4):193–195, 1978.

[For14] Michael Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis, Massachusetts Institute of Technology, 2014.

[FS13] Michael A. Forbes and Amir Shpilka. Quasipolynomial-Time Identity Testing of Noncommutative and Read-Once Oblivious Algebraic Branching Programs. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 243–252, 2013. Full version at arXiv:1209.2408.

[FSS14] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 867–875, 2014.

[FSV18] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct Hitting Sets and Barriers to Proving Lower Bounds for Algebraic Circuits. *Theory of Computing*, 14(1):1–45, 2018.

[GKKS14] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *Journal of the ACM*, 61(6):33:1–33:16, 2014. Preliminary version in the *28th Annual IEEE Conference on Computational Complexity (CCC 2013)*. Pre-print available at eccc:TR12-098.

[GKS16] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity Testing for Constant-Width, and Commutative, Read-Once Oblivious ABPs. In *Proceedings of the 31st Annual Computational Complexity Conference (CCC 2016)*, pages 29:1–29:16, 2016. arXiv:1601.08031.

[GKSS17] Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017. Pre-print available at arXiv:1701.01717.

[GKSS19] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from Algebraic Hardness: Treading the Borders. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 147–157. IEEE Computer Society, 2019.

[GKST15] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic Identity Testing for Sum of Read-once Oblivious Arithmetic Branching Programs. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC 2015)*, pages 323–346, 2015. arXiv:1411.7341.

[HS80] Joos Heintz and Claus-Peter Schnorr. Testing Polynomials which Are Easy to Compute (Extended Abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC 1980)*, pages 262–272. ACM, 1980.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[Koi11] Pascal Koiran. Shallow circuits with high-powered inputs. In *Innovations in Computer Science - ICS 2011, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 309–320. Tsinghua University Press, 2011.

[KST19] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 639–646. SIAM, 2019.

[LFKN90] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic Methods for Interactive Proof Systems. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS 1990)*, pages 2–10, 1990.

[LLS17] Guillaume Lagarde, Nutan Limaye, and Srikanth Srinivasan. Lower Bounds and PIT for Non-Commutative Arithmetic circuits with Restricted Parse Trees. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:77, 2017. eccc:TR17-077.

[LMP16] Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. Non-commutative computations: lower bounds and polynomial identity testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:94, 2016. eccc:TR16-094.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. Preliminary version in the *19th Annual ACM Symposium on Theory of Computing (STOC 1987)*.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, pages 410–418, 1991. Available on citeseer:10.1.1.17.5067.

[Ore22] Øystein Ore. Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

[Raz06] Ran Raz. Separation of Multilinear Circuit and Formula Size. *Theory of Computing*, 2(1):121–135, 2006. Preliminary version in the *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*. Pre-print available at eccc:TR04-042.

[Raz10] Ran Raz. Elusive Functions and Lower Bounds for Arithmetic Circuits. *Theory of Computing*, 6(1):135–177, 2010.

[RR97] Alexander A. Razborov and Steven Rudich. Natural Proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. Preliminary version in the *19th Annual IEEE Conference on Computational Complexity (CCC 2004)*.

[Sax08] Nitin Saxena. Diagonal Circuit Identity Testing and Lower Bounds. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, pages 60–71, 2008. Pre-print available at eccc:TR07-124.

[Sch80] Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, 1980.

[Sri19] Srikanth Srinivasan. Strongly Exponential Separation Between Monotone VP and Monotone VNP. *CoRR*, abs/1903.01630, 2019. Pre-print available at arXiv:1903.01630.

[SS94] Michael Shub and Steve Smale. *On the intractability of Hilbert's Nullstellensatz and an algebraic version of "NP= P?"*. 1994.

[ST18] Ramprasad Saptharishi and Anamay Tengse. Quasipolynomial Hitting Sets for Circuits with Restricted Parse Trees. In *38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2018, December 11-13, 2018, Ahmedabad, India*, volume 122 of *LIPIcs*, pages 6:1–6:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[Str73] Volker Strassen. Die Berechnungskomplexität Von Elementarsymmetrischen Funktionen Und Von Interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, 1973.

[Val79] Leslie G. Valiant. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 1979)*, pages 249–261, 1979.

[Yeh19] Amir Yehudayoff. Separating monotone VP and VNP. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 425–429. ACM, 2019.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979.