Guest Column: Parting Thoughts and Parting Shots (Read On for Details on How to Win Valuable Prizes!)[1]

Eric Allender[2]

Abstract

A list of open questions is found herein, each with a bounty of $1000.

1 Introduction

The end is near. More precisely, the end of my time as a (non-emeritus) professor is near. Thus I'm particularly grateful to Lane for inviting me to share some thoughts here, because there are a few things that I want to get off my chest. The Computational Complexity Column provides the perfect podium for this.

I have some scores to settle.

My upcoming retirement in 2023 provides an opportunity to reflect on my career at Rutgers. I've had the chance to interact with some wonderful students and colleagues, and I've been lucky to have my research efforts be appreciated [AM17]. But there have also been disappointments. I'm not referring here to the myopic conference program committees that accepted clearly inferior work while rejecting my submissions; after all, I've been on enough program committees over the years, so that I've been complicit in my own share of myopic decisions. No, I'm referring instead to something much more problematic. I'm referring to the adversaries that have caused me so much grief, that I'm offering a bounty to have them eliminated.

I'm referring to the open problems that seem like they should be solvable, but which remain at large to this day.

It comes as no surprise to me that these problems have defeated *my* best efforts. After all, my best efforts have frequently fallen short. But I'm fortunate to be part of a community of hard-working and brilliant theoreticians, and I've been happy to see them succeed where I have failed. (The list of papers that solve problems that I left as open questions includes [GW93, RRW94, Hir18, Agr11, ABL98, KV10, CDE$^+$14, BTV09, Hes01] and I'm sure that this is incomplete.)

I'm not bothering to offer a bounty for the big open questions that are already well known to everyone who is likely to be reading this column; the payoff for solving those problems is already huge and dwarfs any increment I might be able to offer. Instead, I've curated a selection of problems that I truly believe ought to yield to a properly-mounted offensive. These are problems that I believe

[2] Department of Computer Science, Rutgers University, Piscataway, NJ 08854, USA. `allender@cs.rutgers.edu`. Supported by NSF Grants CCF-1909683 and CCF-1909216.

deserve to be better known. And they're problems that – I hope – will be entertaining to read about.

So, with no further ado:

2 Equity for PP and #P: Why should we have stronger lower bounds for one than for the other?

There has been significant progress in the last few years, resulting in improved circuit lower bounds. In particular, ACC^0 circuits require superpolynomial size in order to recognize certain languages in NQP (nondeterministic quasipolynomial time) [MW20, Che19, CR20]. This is an extremely important line of work, and by some metrics it brings us close to having interesting circuit lower bounds for problems in NP.

But – although NQP is "close" to NP in some sense – the reader should note that NQP is not known (or widely believed) to lie in PSPACE. Thus the lower bounds of [MW20, Che19, CR20] are incomparable with the *uniform* circuit lower bounds that are known for PP, #P, and other classes in the counting hierarchy. Admittedly, uniformity is a strong restriction – but even when we restrict our attention to uniform circuits, there are some interesting open questions. That is the subject of this section.

The circuit classes we'll be concerned with in this section are TC^0 (constant depth threshold circuits) and ACC^0 (constant depth circuits of AND, OR, and MOD_m gates for some fixed modulus m). A circuit family $\{C_n : n \in \mathbb{N}\}$ is *uniform* if there is a (very) efficient way[3] to obtain the description of C_n given n. Any problem that is complete for PSPACE under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions (or even under $\leq_{\mathrm{m}}^{\mathsf{TC}^0}$ reductions) requires uniform TC^0 circuits of size at least 2^{n^ϵ} for some $\epsilon > 0$ [All99] (and this size bound cannot be improved, since it is easy to see via a padding argument that, for every $\epsilon > 0$ there are complete sets for PSPACE that have uniform AC^0 circuits of size 2^{n^ϵ}). Despite some effort, however, there is still no example of a subclass of PSPACE that has been shown to require exponential-size uniform TC^0 circuits. Instead, there are results that show only $\frac{1}{k}$-exponential size bounds, using the terminology that was introduced by Miltersen, Vinodchandran, and Watanabe [MVW99]. Informally, a function T is said to have at most half-exponential growth if $T(T(n^{O(1)})) = 2^{n^{o(1)}}$, and more generally it is said to have at most $\frac{1}{k}$-exponential growth if the k-fold composition $T(T(\ldots T(n)\ldots)) = 2^{o(n)}$. If T is less than $\frac{1}{k}$-exponential, for every k, then no set that is complete for PP under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ has uniform TC^0 circuits of size $T(n)$ [All99]. I do not think that this is optimal, but I'm not offering a bounty for improving that lower bound. There has already been work extending and building on [All99] in various ways, without obtaining larger bounds (see [KP09, Kin12]). Instead, I want to call attention to a different, more glaringly deficient lower bound.

Prior to the $\frac{1}{k}$-exponential uniform TC^0 lower bound for PP that was presented in [All99], somewhat similar lower bounds against uniform ACC^0 circuits were presented in [AG94]. There, it was shown that there are problems in PP that cannot be computed by uniform ACC^0 circuits whose growth rate is at most half-exponential. Thus, the size lower bound is larger than in the lower bound of [All99], which makes a certain amount of sense, since the class of circuits is weaker. But [AG94] also shows that the Permanent (and any problem complete for #P under AC^0 reductions)

[3] The reader is advised to consult the papers cited in this section for more details on the type of "uniformity" that is considered here, along with the motivation for considering this particular notion.

requires ACC^0 circuit size at least 2^{n^ϵ} for some $\epsilon > 0$. That is, the uniform ACC^0 circuit size lower bounds for $\#\mathsf{P}$ are *much, much* stronger than the bounds that we have for PP.

Does that seem reasonable? We are accustomed to think that $\#\mathsf{P}$ and PP have similar complexity ..., but the standard reductions of $\#\mathsf{P}$ problems to those in PP rely on binary search, which is unlikely to be possible using ACC^0 circuits. Still, there is a history of successful attempts to prove exponential bounds, where the naïve approach would yield a half-exponential bound [AGHK11, TV07].

I see no reason why there should not be a simple proof, showing that complete sets for PP need to have ACC^0 circuits that are as large as those that are required by $\#\mathsf{P}$-complete problems. I think that such a proof would likely yield techniques that will be useful in other settings. Thus I offer the following bounty:

Wanted: \$1000 REWARD! Open Question 1 *Do problems that are complete for* PP *under* $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ *reductions require uniform* ACC^0 *circuits of size* 2^{n^ϵ} *for some* $\epsilon > 0$*?*

Note that it suffices to prove that this lower bound holds for some "standard" PP-complete problem, such as determining if a given Boolean formula (or circuit) is satisfied by at least half of all possible assignments.

3 Uniformity and the Prime Numbers

How hard is it, to tell if a number is prime?

One way answer to this question is to provide an *upper* bound. The set of prime numbers lies in P [AKS04], and any improvement in this upper bound would be extremely interesting. But that's not the question I want to focus on here.

Instead, I'm interested in improved *lower bounds* on the complexity of the set of prime numbers. Currently, the best lower bounds that we have for the set of prime numbers follow from the fact that the set is hard for TC^0 under nonuniform $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reductions [ASS01]. I'd be very interested in learning that the set of prime numbers is hard for an even larger subclass of P – but I'm not going to offer a bounty for that, because I'm not very confident that the set of primes really is hard for any class larger than TC^0.

Similarly, I'd be very interested in knowing if the set of primes is hard for TC^0 under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions – but again, I'm not going to offer a bounty for that improvement. It's not clear to me that there should even be an $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reduction from PARITY to the set of primes. In fact, it might even be feasible to *prove* that the set of primes is not complete under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions for P or NL or any other familiar complexity class.[4]

Instead, I want to draw your attention to the question of uniformity. The nonuniform reduction in [ASS01] is quite simple; the idea can be illustrated by considering how to reduce the MOD_3 problem to the set of primes. First, we observe that MOD_3 reduces to the question of whether a given number (in binary notation) is a multiple of 3. (This is a simple uniform reduction.) Next, we note that if y is a multiple of 3, then for every number r, $y + 3r$ is also a multiple of 3, whereas if y is not a multiple of 3, then $y + 3r$ is reasonably likely to be prime (for randomly-chosen r). This forms the basis of a (uniform) probabilistic reduction to the set of primes, which can then

[4] I discussed some ideas about how to attack problems like this in [All01]; some further reflections on whether this approach might be difficult can be found in [All14].

be turned into a non-uniform deterministic reduction. To date, there is still no known uniform $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reduction of MOD_3 to the set of primes. Stated another way: We know that there are AC^0 circuits that reduce the question "*Is y a multiple of 3?*" to the question "*Is y composite?*" – but we don't know how to build those circuits. This seems unreasonable. The first question is very easy to answer, and seems (in some sense) to be a special case of the second question. Thus I offer the following bounty:

Wanted: \$1000 REWARD! Open Question 2 *Is the set of prime numbers hard for* TC^0 *under* $\mathsf{DLOGTIME}$*-uniform* $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ *reductions?*

In general, I think that the following heuristic offers good guidance:

> *If there is a non-uniform (or not-very-uniform) circuit family accomplishing some task, very likely there is a* $\mathsf{DLOGTIME}$*-uniform circuit family for the same task.*

Here is a list of some examples from history, to illustrate how this heuristic usually works out:

- Agrawal's Isomorphism Theorem [Agr11] (showing that, for most complexity classes of interest, the sets complete under uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions are all AC^0-isomorphic) was initially known only in the non-uniform setting [AAR98].

- The P-uniform TC^0 division algorithm of Beame, Cook, and Hoover [BCH86] was later replaced by a $\mathsf{DLOGTIME}$-uniform algorithm [HAB02].

- Torán showed that Graph Isomorphism is hard for DET under logspace-uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, and it was later shown to be complete under $\mathsf{DLOGTIME}$-uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions [All04b]. (Other related examples are also discussed in [All04b].)

- Frandsen, Valence, and Barrington [FVB94] introduced a class of functions called AE, which they showed is sandwiched between $\mathsf{DLOGTIME}$-uniform TC^0 and P-uniform TC^0. Later, Healy and Viola showed that AE coincides with $\mathsf{DLOGTIME}$-uniform TC^0 [HV06].

Of course, there are also some counterexamples. It was shown in [ABK$^+$06b] that the set of strings with high space-bounded Kolmogorov complexity is complete for PSPACE under non-uniform nonadaptive TC^0 reductions, but that this fails even under L-uniformity. Still, it strikes me as unlikely that nonuniformity is essential in order to reduce TC^0 to the set of prime numbers.

4 ACC^0 Again: A Retraction

This section is the most difficult for me to write.

Hansen gave a surprising and beautiful characterization of ACC^0 as the class of problems having *planar* circuits of constant width [Han06]. Later, an extension of this result was published [ADR05], claiming to show that this characterization holds not only for circuit families embeddable on a surface of genus zero, but even for circuit families $\{C_n : n \in \mathbb{N}\}$ where C_n can be embedded on a surface of genus $\log^{O(1)} n$.

The main result of [ADR05] may be true, but the argument presented in the paper is incorrect.[5]

[5] Without going into too much detail, in the proof of [ADR05, Lemma 8], the argument breaks down at the statement: "Thus we can view them as being stripes arranged along the sides of the cylinder. In this way, each handle connection h_i has an East neighbor and a West neighbor…" There is a counterexample to this assertion.

We were blissfully unaware of the bug until we were revising the article for submission to a journal. It seemed easy to fix at first. And then we found a bug in the fix, which also seemed easy to fix. And so on... At this point, it is becoming clear that the bug is not going to be fixed before my impending retirement. The claim is definitely true for genus 1, and probably true for genus $O(1)$. Perhaps a fresh approach (with fresh eyes) will settle the matter?

Wanted: $1000 REWARD! Open Question 3 *Does every language accepted by constant-width circuit families of polylogarithmic genus lie in* ACC^0*?*

5 Ambiguously Unique

The complexity class UP (sometimes known as "unique" P or "unambiguous" P) is familiar to most people who are likely to read this article. A language is in UP if it is accepted by some NP machine that never has more than one accepting computation path. Thus if a string is accepted, there is a *unique* witness for acceptance, and there is no *ambiguity* about how to prove that x is accepted. There are several different ways that this can be formalized.

When it comes to defining a "unique" or "unambiguous" analog of UP as a subclass of NL, it turns out that some of the "several different ways" to formalize this notion yield classes that appear not to coincide. Buntrock *et al.* [BJLR91] identified *three* different classes, which have come to be known as StUL, RUL and UL, respectively.

Most probably, UL is just another name for NL. In particular, if there is any set in $\mathsf{DSPACE}(n)$ that requires exponential-size circuits, then $\mathsf{UL} = \mathsf{NL}$ [ARZ99], and (unconditionally) they coincide in the setting of nonuniform complexity [RA00].[6] Thus anything that is true for NL should also be true for UL. Which brings us to the first bounty in this section:

Wanted: $1000 REWARD! Open Question 4 *Is* UL *closed under complement?*

Although UL is not yet known to be equal to NL, it is known to contain some interesting problems, such as the reachability problem for directed planar graphs [BTV09, TV12] (see also [KV10, GST19, DGJ$^+$21]) and some other graph problems [TW14, LMN10]. Also, there is a problem that is (unconditionally) in UL that is hard for NL under (nonuniform) projections. All of those problems are also known to be in coUL. Thus we have *no* good "candidates" for being in UL and not in coUL. Yet there is still no proof that $\mathsf{UL} = \mathsf{coUL}$. One way to solve this problem, of course, would be to show that $\mathsf{UL} = \mathsf{NL}$, but it might be considerably easier to show directly that UL is closed under complement.

If UL were known to have a complete set under $\leq_{\mathrm{m}}^{\mathsf{L}}$ reductions, then it would suffice to show that this problem is also in coUL, in order to show that $\mathsf{UL} = \mathsf{coUL}$. However, no such complete set is known to exist. The situation is quite different for RUL (defined in terms of NL machines that have at most one path between any two reachable[7] configurations). RUL has a complete set [Lan97], and RUL is closed under complement [BJLR91]. RUL is also a fairly "robust" class; it was shown to coincide with a class that allows a polynomial amount of "ambiguity" [GSTV14]. In contrast to UL, there is little reason to believe that $\mathsf{RUL} = \mathsf{NL}$. In fact, a better upper bound is known for the space complexity of problems in RUL: $(\log^2 n)/\log\log n$ [AL98], which is an improvement over the

[6] For the best current simulation of NL via unambiguous logspace, see [vMP19].

[7] The name RUL comes from "Reach"-UL, because of this restriction to reachable configurations.

$\log^2 n$ space bound for NL from Savitch's theorem. I suspect that this space bound for RUL can be improved.

I'm even more confident that an improved space bound can be proved for StUL, or "Strongly-Unambiguous Logspace" (defined in terms of NL machines that have at most one path between *any* two configurations, regardless of whether they are reachable). The configuration graphs of StUL machines are known as *mangroves*, which have turned out to be useful in other investigations of logspace computation [EJT10]. There is no language known to reside in StUL that is not also known to reside in L. I suspect that $\mathsf{StUL} = \mathsf{L}$, but I would be happy to see a proof of something much weaker:

Wanted: $1000 REWARD! Open Question 5 *Is* $\mathsf{StUL} \subseteq \mathsf{DSPACE}(o((\log^2 n)/\log\log n))$*?*

6 How Robust Is the Determinant?

The previous section dealt with NL, which belongs on everyone's short list of important complexity classes, because it captures the complexity of a great many computational problems that we care about (such as the problem of finding shortest paths in graphs). In this section, we consider the related class $\#\mathsf{L}$. Although $\#\mathsf{L}$ also captures the complexity of some natural problems (such as counting the number of s-t paths in a DAG), the strongest argument for being interested in $\#\mathsf{L}$ is this: It captures the complexity of computing the determinant. More precisely: GapL is the class of functions that can be represented as the difference of two $\#\mathsf{L}$ functions. The determinant of integer matrices is complete for GapL under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions. That is, for any $f, g \in \#\mathsf{L}$ there is an AC^0 function h such that $f(x) - g(x) = \det(h(x))$ [Dam91, Tod91, Vin91, MV97].

There are many reasons why complexity theoreticians are interested in the determinant. It plays a central role in the theory of algebraic circuits. Also, there are several Boolean complexity classes that are intimately connected to the determinant:

- $\mathsf{C}_=\mathsf{L} = \{A : \exists f \in \mathsf{GapL}\ (x \in A \Leftrightarrow f(x) = 0)\}$. The set of singular matrices is complete for $\mathsf{C}_=\mathsf{L}$ (essentially by definition).
- $\mathsf{L}^{\mathsf{C}_=\mathsf{L}}$ is the class of languages $\leq_{\mathrm{T}}^{\mathsf{L}}$ reducible to a language in $\mathsf{C}_=\mathsf{L}$. Problems complete for $\mathsf{L}^{\mathsf{C}_=\mathsf{L}}$ under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions include computing the rank of a matrix, and solving systems of linear equations [ABO99].
- $\mathsf{PL} = \{A : \exists f \in \mathsf{GapL}\ (x \in A \Leftrightarrow f(x) > 0)\}$. (This is the "unbounded error" version of probabilistic logspace originally defined by Gill [Gil77].) The set of matrices with positive determinant is complete for PL.

Remark 1 *It may well be true that* $\mathsf{C}_=\mathsf{L}$ *is closed under complement. If this is the case, then* $\mathsf{C}_=\mathsf{L} = \mathsf{L}^{\mathsf{C}_=\mathsf{L}}$ *[ABO99].*

A recurring theme in complexity theory is that "natural" problems that are complete for some complexity class under one notion of reducibility (such as $\leq_{\mathrm{m}}^{\mathsf{L}}$ reductions) usually remain complete under very restrictive reductions (such as $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions, or even projections). For problems that are reducible to their complement, even using very powerful notions of reducibility (such as various types of Turing reducibility) yield the same class of problems. For example, if we let stconn

denote the standard complete problem for NL, then the set of problems reducible to stconn is NL, regardless[8] of whether "reducible" means "under $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions" or "under $\leq_{\mathrm{T}}^{\mathsf{NC}^1}$ reductions".

The main open question in this section is whether the determinant shares this "robustness" property. When Cook defined DET as the class of problems reducible to the determinant, he defined it in terms of $\leq_{\mathrm{T}}^{\mathsf{NC}^1}$ reductions [Coo85]. (Equivalently: $\mathsf{DET} = \mathsf{NC}^1(\#\mathsf{L})$.) But would it have made a difference, if he had defined it in terms of $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reductions? That is: Is $\mathsf{AC}^0(\#\mathsf{L}) = \mathsf{NC}^1(\#\mathsf{L})$? There are some reasons to suspect that they might be equal. For instance:

- $\mathsf{L}^{\mathsf{C}_=\mathsf{L}} = \mathsf{AC}^0(\mathsf{C}_=\mathsf{L}) = \mathsf{NC}^1(\mathsf{C}_=\mathsf{L})$ [ABO99].
- $\mathsf{PL} = \mathsf{AC}^0(\mathsf{PL}) = \mathsf{NC}^1(\mathsf{PL})$ [BF00].

Wanted: $1000 REWARD! Open Question 6 *Is it the case that*

$$\mathsf{DET} = \mathsf{NC}^1(\#\mathsf{L}) = \mathsf{AC}^0(\#\mathsf{L})?$$

The reader may want to consult [All04a, AAM03] for some additional discussion of this question.

7 Boolean vs. Arithmetic Formulae

I am predisposed to believe that, someday, we will really understand the framework of complexity classes, and that the final picture will be *beautiful*. In particular, it should not be too cluttered with jarring oddities that spoil the picture. This section is all about tidying up the clutter.

Readers of this article are likely to be quite familiar with NC^1: the class of problems that have families of Boolean formulae of polynomial size. You might be less familiar with the "counting" version: $\#\mathsf{NC}^1$. There are several different equivalent ways to define $\#\mathsf{NC}^1$; perhaps the simplest one is in terms of families of arithmetic circuits (with + and × gates of fan-in two, evaluated over $\mathbb{N}$) of polynomial size and depth $O(\log n)$.

There is a simple argument, due to Jung [Jun85], showing that every function in $\#\mathsf{NC}^1$ is computed by Boolean circuits[9] of polynomial size, fan-in two, and depth $O(\log n \log^* n)$. In other words, $\#\mathsf{NC}^1$ and NC^1 are very nearly the same. Furthermore, there are at least five other complexity classes that have received study, that would all collapse to NC^1 if $\#\mathsf{NC}^1 = \mathsf{NC}^1$ [CMTV98, DMR$^+$12, All04a, AKM19]. Thus it would certainly "tidy things up" if one could show that NC^1 and $\#\mathsf{NC}^1$ coincide, improving Jung's $O(\log n \log^* n)$ upper bound that has stood since 1985. But I'm willing to pay full price for something far more modest: Give *any* improvement.

Wanted: $1000 REWARD! Open Question 7 *Is it the case that every function in $\#\mathsf{NC}^1$ has Boolean circuits of fan-in two, polynomial size and depth $o(\log n \log^* n)$?*

[8] This may be an appropriate time to mention something counterintuitive about $\leq_{\mathrm{T}}^{\mathsf{L}}$ and $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$ reductions. Although $\mathsf{AC}^0 \subsetneq \mathsf{NC}^1 \subseteq \mathsf{L}$, it is nonetheless the case that $B \leq_{\mathrm{T}}^{\mathsf{L}} A$ implies $B \leq_{\mathrm{T}}^{\mathsf{AC}^0} A$, for problems A that are hard for NL. Thus, for instance, every problem in $\mathsf{L}^{\mathsf{C}_=\mathsf{L}}$ is $\leq_{\mathrm{T}}^{\mathsf{AC}^0}$-reducible to the set of singular matrices [AO96].

[9] I provide an arguably simpler presentation of Jung's proof in [All04a].

8 Reducing the Degree

In this section, we investigate another aspect of the "de-cluttering" program that was introduced in the previous section. We will focus on the notion of the "degree" of a Boolean (or arithmetic) circuit. Skyum and Valiant [SV85] may have been the first to identify the degree of a Boolean circuit as an important consideration in complexity theory. (See also [Coo85].) The class of languages accepted by uniform circuit families of polynomial size and polynomial degree is known by several names:

- SAC^1 (Log-depth circuits with unbounded fan-in OR gates, and bounded fan-in AND gates; known as semi-unbounded fan-in circuits.)
- LogCFL (The class of languages $\leq_m^{\mathsf{L}}$-reducible to context-free languages.)
- $\mathsf{NAuxPDA}(\log, n^{O(1)})$ (The class of languages accepted by nondeterministic auxiliary pushdown automata in polynomial time and logarithmic workspace.)

The reader can consult the textbook by Vollmer [Vol99] for more background on this topic.

Of course, the notion of polynomially-bounded degree also plays a central role in the theory of algebraic computation, where the complexity class VP denotes the class of families polynomials that can be represented by circuits of polynomial size and polynomial degree. As in the Boolean case, VP corresponds to polynomial-size semi-unbounded circuits of logarithmic depth (with unbounded fan-in $+$ and bounded fan-in $\times$ gates) [Vin91, AJMV98]. The auxiliary pushdown automaton model of computation has also been useful in working with VP, as with the Boolean case (see, e.g., [AJMV98, Men13]).

Having interesting complete problems helps motivate interest in a complexity class. However, VP was studied intensely for decades before finally accumulating a collection of interesting natural complete problems [Men13, DMM$^+$16, MS18, CLP21, CLV21]. I recommend the excellent survey by Mahajan [Mah14] for a discussion of these developments, as well as for an interesting perspective on algebraic computation.

Our focus in this section will not be on VP *per se*, but rather on a Boolean class that corresponds in a natural way to VP. If we are working over $\mathbb{F}_2$, then given any polynomial p in n variables and a string x of length n representing an assignment to those variables, $p(x)$ takes on a value in $\{0, 1\}$, and in this way any family in VP naturally corresponds to a language. In [AGM17], this class was denoted $\mathsf{VP}(\mathbb{F}_2)$. As with LogCFL, this class has many equivalent names:

- $\mathsf{VP}(\mathbb{F}_2)$
- $\oplus\mathsf{AuxPDA}(\log, n^{O(1)})$ (The class of languages L for which there is a nondeterministic auxiliary pushdown automaton M running in polynomial time and logarithmic workspace, where $x \in L$ iff M has an odd number of accepting computations on x.)
- $\mathsf{SAC}^1[\oplus, \wedge]$ (Log-depth circuits with unbounded fan-in PARITY gates and bounded fan-in AND gates.)

$\mathsf{VP}(\mathbb{F}_2)$ has a number of natural complete problems, inherited from the complete polynomials for VP.

In contrast, the other two complexity classes that we will focus on in this section really don't have any interesting natural complete problems of which I am aware. Those classes are:

- AC^1 (Log-depth circuits of unbounded fan-in AND and OR gates.)
- $\mathsf{AC}^1[\oplus]$ (Just like AC^1, but now with unbounded fan-in PARITY gates as well.)

The known relationships among the classes mentioned in this section are:

$$\mathsf{SAC}^1 \subseteq \mathsf{AC}^1 \subseteq \mathsf{AC}^1[\oplus].$$

$$\mathsf{VP}(\mathbb{F}_2) \subseteq \mathsf{AC}^1[\oplus].$$

In addition, under a plausible derandomization hypothesis, $\mathsf{SAC}^1 \subseteq \mathsf{VP}(\mathbb{F}_2)$ [GW96, RA00, ARZ99].

It is essentially obvious that $\mathsf{AC}^1[\oplus]$ corresponds to languages represented by algebraic circuits over $\mathbb{F}_2$ of polynomial size, logarithmic depth, and degree $n^{O(\log n)}$. Somewhat surprisingly, it is shown in [AGM17] that $\mathsf{AC}^1[\oplus]$ *also* corresponds to languages represented by algebraic circuits over $\mathbb{F}_2$ of polynomial size, logarithmic depth, *and degree* $n^{O(\log\log n)}$! That is, the degree can be *reduced* significantly. The proof in [AGM17] is not difficult, and I very much doubt that the degree bound $n^{O(\log\log n)}$ is optimal.

The obvious question is:

> *Does this degree collapse (from quasipolynomial to $n^{O(\log\log n)}$) go further? Does it go all the way to $n^{O(1)}$? Equivalently: is $\mathsf{AC}^1[\oplus]$ equal to $\mathsf{VP}(\mathbb{F}_2)$?*

This would be asking quite a lot, since $\mathsf{VP}(\mathbb{F}_2)$ is not even known to contain NL (although, under a plausible derandomization hypothesis, it contains even the seemingly-larger class LogCFL, as mentioned above). Thus, I'm going to offer full payment for a weaker result:

Wanted: $1000 REWARD! Open Question 8 *Is AC^1 contained in $\mathsf{VP}(\mathbb{F}_2)$ under a plausible derandomization hypothesis?*

9 Numbers as Easy as π

A language (say, $A \subseteq \{0,1\}^* = \{s_0, s_1, \ldots\}$ where $s_0 = \lambda$ (the empty string)) can be equated with its characteristic sequence $\chi_A = b_0 b_1 b_2 \ldots$ where $b_i = 1$ if $s_i \in A$ and $b_i = 0$ otherwise. But the sequence χ_A is also the binary representation of a real number in the interval $[0,1]$. For instance, the sequences $1000\ldots$ and $0111\ldots$ (corresponding to the languages $\{\lambda\}$ and $\{x : x \neq \lambda\}$, respectively) both denote the number $\frac{1}{2} = \sum_{i=2}^{\infty} 2^{-i}$. More generally, the finite and co-finite languages correspond exactly to dyadic rational numbers. Any real number in $[0,1]$ that is *not* a dyadic rational has exactly one binary representation, and hence corresponds to exactly one language.

So, what is the complexity of (the fractional part of) π? It lies in $\mathsf{PH}^{\mathsf{CH}_3}$ [ABDP23] (improving a PSPACE upper bound that follows from [Yap10]). (Here, CH_k refers to the k^{th} level of the counting hierarchy $\mathsf{PP}, \mathsf{PP}^{\mathsf{PP}}$, etc., and PH refers to the polynomial hierarchy.) How about e? The same paper gives an upper bound of $\mathsf{PH}^{\mathsf{CH}_4}$. Is there any reason why e should be more difficult than π?

...But that's not the question I want to call your attention to here.

Every algebraic number lies in $\mathsf{PH}^{\mathsf{CH}_3}$ [ABDP23], improving on a bound that is implicit in the work of Jeřábek [Jeř12]. Algebraic numbers come in two flavors: rational and irrational. The rational numbers all all lie in ACC^0 [BC91], and in fact they all are regular sets. Furthermore, for every odd modulus m there is a rational number α_m that is hard for $\mathsf{AC}^0[m]$ under projections, and hence lies outside of $\mathsf{AC}^0[p]$ for any prime p that doesn't divide m [ABDP23].

It was a fairly big result a few years ago, when Adamczewski, Bugeaud, and Luca proved that no irrational algebraic number is regular [ABL04, AB07]. Thus any regular language that does not correspond to a rational number (i.e., any regular language whose characteristic sequence is not ultimately periodic) corresponds to a transcendental real number. Freivalds has written an excellent survey explaining some of the reasons that motivate a study of the "complexity" of real numbers in this setting [Fre12].

What about the irrational algebraic numbers? Although it's reasonable to conjecture that they're at least as difficult as the rational ones, we don't know whether that's the case.

Wanted: \$1000 REWARD! Open Question 9 *Is every irrational algebraic number in* AC^0*? Is there any irrational algebraic number in* PH*? (Full payment for answering either of these questions.)*

I think that it would be instructive to see a proof that some irrational algebraic number lies outside of AC^0 (or even, that this holds for *every* irrational algebraic number). It would be remarkable if the irrational algebraic numbers yielded a class of languages in CH that are not in PH. It is clear that any argument showing that some irrational algebraic number lies in PH will have to make use of very different techniques than were used in [ABDP23].

10 The Random Strings

The end is near. In this final section, we pose a question that lies at the intersection of complexity theory and computability theory.

The story for this section begins with this paper: [ABK$^+$06b]. When we began work on this project, our focus was on resource-bounded Kolmogorov complexity, and it's accurate to say that the final paper still is primarily on that topic. But we also proved some results about the *undecidable* set of Kolmogorov-random strings.[10] Namely, if we let R denote the set of Kolmogorov-random strings, then R is hard for PSPACE under $\leq_{\mathrm{T}}^{\mathsf{P}}$ reductions, and *every* computably-enumerable set (including the Halting Problem) is reducible to R under $\leq_{\mathrm{T}}^{\mathsf{P/poly}}$ reductions.

Later, additional results with this flavor were proved:

- $\mathsf{BPP} \subseteq \mathsf{P}_{tt}^{R}$ [BFKL10]. (I.e., every problem in BPP is reducible to R via a nonadaptive polynomial-time reduction.)
- $\mathsf{NEXP} \subseteq \mathsf{NP}^{R}$ [ABK06a].

These results were fairly "robust", in that they held for essentially all of the most common ways to define "Kolmogorov-random" (for example, in terms of "plain Kolmogorov complexity" or "prefix-free Kolmogorov complexity"). And – it almost goes without saying – they also held, no matter which "universal Turing machine" was being used, to define Kolmogorov complexity.

Remark 2 *…But there seemed to be something odd about these results. Did it even make sense to talk about efficient reductions to a non-computable language? Were these results trivial? Is it perhaps the case that* all *computably-enumerable languages are in* P^{R}*?*

[10] Readers who are unfamiliar with Kolmogorov complexity may want to consult some of the excellent texts on this topic, such as [LV19, DH10].

In order to move on to the next chapter in this narrative, it is necessary to start being more specific about the ways to define the "Kolmogorov-random strings". Define R_{C_U} to be the set of strings x with "plain" Kolmogorov complexity $C_U(x) \geq |x|$ where U is the universal Turing machine being used to define plain Kolmogorov complexity. That is, $R_{C_U} = \{x : \neg\exists d \ |d| < |x|, U(d) = x\}$. R_{K_U} is defined similarly, in terms of the version of prefix-free Kolmogorov complexity defined using universal Turing machine U. This additional notation is needed, in order to answer the questions raised in Remark 2:

Theorem 1 *[AFG13, CDE⁺14]*

- $\mathsf{BPP} \subseteq \bigcap_U \mathsf{P}_{tt}^{R_{K_U}} \subseteq \mathsf{PSPACE} \subseteq \bigcap_U \mathsf{P}^{R_{K_U}}$.
- $\mathsf{NEXP} \subseteq \bigcap_U \mathsf{NP}^{R_{K_U}} \subseteq \mathsf{EXPSPACE}$.

In other words, the class of languages that are efficiently reducible to the Kolmogorov-random strings in the prefix-free setting, regardless of the universal machine that is being used to define Kolmogorov complexity, *is a complexity class.*

I made some conjectures about just which complexity classes can be characterized in this way [All12], which turned out to be spectacularly wrong. Hirahara [Hir20] has done some outstanding work shedding more light on what the true answer is, including showing that $\mathsf{EXP}^{\mathsf{NP}} \subseteq \mathsf{P}^R$. Note that this is a very significant improvement over the PSPACE hardness result of [ABK⁺06b]. Still, an exact answer is still not known. I think that this is a very interesting and important direction to pursue. I also think that it is quite worthwhile to understand what is reducible to the Kolmogorov-random strings via nonuniform projections (and other restrictive nonuniform reductions); some initial steps in this direction have recently been taken [AGHR21]. But I'm not offering money for the resolution of those problems.

Instead, I want to direct your attention to the question about what can be reduced to R_{C_U}. In contrast to the situation with R_{K_U}, each set R_{C_U} is hard for the computably-enumerable sets under time-bounded reductions. Let H denote the halting problem (which is complete for the computably-enumerable sets under uniform $\leq_{\mathrm{m}}^{\mathsf{AC}^0}$ reductions); Kummer [Kum96] showed that, for each universal machine U, there is a time-bounded disjunctive truth-table reduction from H to R_{K_U}. (That is, there is a computable function that takes x as input, and produces a list of strings, with the property that $x \in H$ if and only if at least one of the strings is in R_{C_U}.) However, it was shown in [ABK06a] that, no matter what computable time bound t one picks, there is some U such that the disjunctive truth-table reduction from H to R_{K_U} requires more time than t. It is also shown in [ABK06a] that there is some U such that the reduction from H to R_{K_U} can be accomplished in doubly-exponential time, but it is not known if this bound can be improved, for any U.

These results from [ABK06a] are only for disjunctive-truth-table reductions. *Nothing* is known about $\leq_{\mathrm{T}}^{\mathsf{P}}$ reductions.

Wanted: $1000 REWARD! Open Question 10 *Is $H \in \mathsf{P}^{R_{C_U}}$ for some universal machine U? Is $H \notin \mathsf{P}^{R_{C_U}}$ for some universal machine U? (Full payment for answering either of these questions.)*

11 Is This Just a Cheap Stunt?

Well, it's definitely a stunt. And it's not a particularly original stunt. A fairly long list of open problems with monetary rewards can be found via an internet search. (See, e.g. [Exc12].) I agree that it's certainly not clear that putting a price on the head of an open problem decreases the time to solution, although I do hope that it might help. After all, there might be a few people who will have read this far, just to see what I'm willing to pay for. And some of those people may have some good ideas.

At least, I hope that we can agree that it's not a particularly *cheap* stunt!

12 The Fine Print

All of these offers are time-limited. The expiration date is no later than my own: my estate will not have any provisions for making these payments after I am dead, so start working now! Even prior to my death, if I am no longer competent to verify claims that you have answered one of these problems, then the offer is no longer in effect. Also, there is some slight chance that I'm already so far into my dotage, that some of these problems have already been solved without my being aware of it. In that case, no reward can be claimed; these cash rewards are only for solutions that are first made public *after* March, 2023 (when this article is scheduled to appear in SIGACT News). Also, lest there be any ambiguity concerning the currency: All bounties are in the amount of 1,000 US Dollars.

Acknowledgments

I thank Lane Hemaspaandra for inviting me to provide a column, and for his comments and corrections. I also thank Pierre McKenzie, Bill Gasarch and Ryan Williams for their helpful suggestions.

References

[AAM03] Eric Allender, V Arvind, and Meena Mahajan. Arithmetic complexity, Kleene closure, and formal power series. *Theory of Computing Systems*, 36(4):303–328, 2003.

[AAR98] Manindra Agrawal, Eric Allender, and Steven Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *Journal of Computer and System Sciences*, 57(2):127–143, 1998.

[AB07] Boris Adamczewski and Yann Bugeaud. On the complexity of algebraic numbers I. Expansions in integer bases. *Annals of Mathematics*, pages 547–565, 2007.

[ABDP23] Eric Allender, Nikhil Balaji, Samir Datta, and Rameshwar Pratap. On the complexity of algebraic numbers, and the bit-complexity of straight-line programs. *Computability (The Journal of the Association Computability in Europe)*, 2023. To Appear. See also ECCC Report TR22-053.

[ABK06a] Eric Allender, Harry Buhrman, and Michal Koucký. What can be efficiently reduced to the Kolmogorov-random strings? *Ann. Pure Appl. Log.*, 138(1-3):2–19, 2006.

[ABK$^+$06b] Eric Allender, Harry Buhrman, Michal Kouckỳ, Dieter Van Melkebeek, and Detlef Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35(6):1467–1493, 2006.

[ABL98] Andris Ambainis, David A. Mix Barrington, and Huong LeThanh. On counting AC^0 circuits with negative constants. In *Proc. 23rd Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 1450 of *Lecture Notes in Computer Science*, pages 409–417. Springer, 1998.

[ABL04] Boris Adamczewski, Yann Bugeaud, and Florian Luca. Sur la complexité des nombres algébriques. *Comptes Rendus Mathematique*, 339(1):11–14, 2004.

[ABO99] Eric Allender, Robert Beals, and Mitsunori Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8(2):99–126, 1999.

[ADR05] Eric Allender, Samir Datta, and Sambuddha Roy. Topology inside NC^1. In *Proc. 20th Annual IEEE Conference on Computational Complexity (CCC)*, pages 298–307. IEEE Computer Society, 2005.

[AFG13] Eric Allender, Luke Friedman, and William I. Gasarch. Limits on the computational power of random strings. *Information and Computation*, 222:80–92, 2013.

[AG94] Eric Allender and Vivek Gore. A uniform circuit lower bound for the Permanent. *SIAM Journal on Computing*, 23(5):1026–1049, 1994.

[AGHK11] Baris Aydinlioglu, Dan Gutfreund, John M. Hitchcock, and Akinori Kawachi. Derandomizing Arthur-Merlin games and approximate counting implies exponential-size lower bounds. *Computational Complexity*, 20(2):329–366, 2011.

[AGHR21] Eric Allender, John Gouwar, Shuichi Hirahara, and Caleb Robelle. Cryptographic hardness under projections for time-bounded Kolmogorov complexity. In *Proc. 32nd International Symposium on Algorithms and Computation (ISAAC)*, volume 212 of *LIPIcs*, pages 54:1–54:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[AGM17] Eric Allender, Anna Gál, and Ian Mertz. Dual VP classes. *Computational Complexity*, 26(3):583–625, 2017.

[Agr11] Manindra Agrawal. The isomorphism conjecture for constant depth reductions. *Journal of Computer and System Sciences*, 77(1):3–13, 2011.

[AJMV98] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-commutative arithmetic circuits: Depth reduction and size lower bounds. *Theoretical Computer Science*, 209(1-2):47–86, 1998.

[AKM19] Eric Allender, Andreas Krebs, and Pierre McKenzie. Better complexity bounds for cost register automata. *Theory of Computing Systems*, 63(3):367–385, 2019.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004.

[AL98] Eric Allender and Klaus-Jörn Lange. RUSPACE($\log n$) $\subseteq$ DSPACE($\log^2 n$ / $\log\log n$). *Theory of Computing Systems*, 31(5):539–550, 1998.

[All99] Eric Allender. The permanent requires large uniform threshold circuits. *Chicago J. Theor. Comput. Sci.*, 1999, 1999.

[All01] Eric Allender. Some pointed questions concerning asymptotic lower bounds, and news from the isomorphism front. In Gheorghe Paun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Current Trends in Theoretical Computer Science, Entering the 21th Century*, pages 25–41. World Scientific, 2001.

[All04a] Eric Allender. Arithmetic circuits and counting complexity classes. In J. Krajíček, editor, *Complexity of Computations and Proofs*, volume 13 of *Quaderni di Matematica*, pages 33–72. Seconda Università di Napoli, 2004.

[All04b] Eric Allender. The division breakthroughs. In Gheorghe Paun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Current Trends in Theoretical Computer Science, The Challenge of the New Century, Vol. 1: Algorithms and Complexity*, pages 147–164. World Scientific, 2004.

[All12] Eric Allender. Curiouser and curiouser: The link between incompressibility and complexity. In *How the World Computes - Turing Centenary Conference and Proc. 8th Conference on Computability in Europe (CiE)*, volume 7318 of *Lecture Notes in Computer Science*, pages 11–16. Springer, 2012.

[All14] Eric Allender. Investigations concerning the structure of complete sets. In *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, volume 26 of *Progress in Computer Science and Applied Logic*, pages 23–35. 2014.

[AM17] V Arvind and Meena Mahajan. A quest for structure in complexity. *Bulletin of the EATCS*, 123, 2017.

[AO96] Eric Allender and Mitsunori Ogihara. Relationships among PL, #L, and the determinant. *RAIRO Theor. Informatics Appl.*, 30(1):1–21, 1996.

[ARZ99] Eric Allender, Klaus Reinhardt, and Shiyu Zhou. Isolation, matching, and counting: Uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59(2):164–181, 1999.

[ASS01] Eric Allender, Michael E. Saks, and Igor E. Shparlinski. A lower bound for primality. *Journal of Computer and System Sciences*, 62(2):356–366, 2001.

[BC91] David A. Mix Barrington and James C. Corbett. A note on some languages in uniform ACC^0. *Theor. Comput. Sci.*, 78(2):357–362, 1991.

[BCH86] Paul Beame, Stephen A. Cook, and H. James Hoover. Log depth circuits for division and related problems. *SIAM Journal on Computing*, 15(4):994–1003, 1986.

[BF00] Richard Beigel and Bin Fu. Circuits over PP and PL. *Journal of Computer and System Sciences*, 60(2):422–441, 2000.

[BFKL10] Harry Buhrman, Lance Fortnow, Michal Koucký, and Bruno Loff. Derandomizing from random strings. In *Proc. 25th Annual IEEE Conference on Computational Complexity (CCC)*, pages 58–63. IEEE Computer Society, 2010.

[BJLR91] Gerhard Buntrock, Birgit Jenner, Klaus-Jörn Lange, and Peter Rossmanith. Unambiguity and fewness for logarithmic space. In *Proc. 8th International Symposium on Fundamentals of Computation Theory (FCT)*, volume 529 of *Lecture Notes in Computer Science*, pages 168–179. Springer, 1991.

[BTV09] Chris Bourke, Raghunath Tewari, and N. V. Vinodchandran. Directed planar reachability is in unambiguous log-space. *ACM Transactions on Computation Theory*, 1(1):4:1–4:17, 2009.

[CDE^{+}14] Mingzhong Cai, Rodney G. Downey, Rachel Epstein, Steffen Lempp, and Joseph S. Miller. Random strings and tt-degrees of Turing complete C.E. sets. *Log. Methods Comput. Sci.*, 10(3), 2014.

[Che19] Lijie Chen. Non-deterministic quasi-polynomial time is average-case hard for ACC circuits. In *60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1281–1304. IEEE Computer Society, 2019.

[CLP21] Prasad Chaugule, Nutan Limaye, and Shourya Pandey. Variants of the determinant polynomial and the VP-completeness. In *Proc. 16th International Computer Science Symposium in Russia (CSR)*, volume 12730 of *Lecture Notes in Computer Science*, pages 31–55. Springer, 2021.

[CLV21] Prasad Chaugule, Nutan Limaye, and Aditya Varre. Variants of homomorphism polynomials complete for algebraic complexity classes. *ACM Transactions on Computation Theory*, 13(4):21:1–21:26, 2021.

[CMTV98] Hervé Caussinus, Pierre McKenzie, Denis Thérien, and Heribert Vollmer. Nondeterministic NC^1 computation. *Journal of Computer and System Sciences*, 57(2):200–212, 1998.

[Coo85] Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1-3):2–21, 1985.

[CR20] Lijie Chen and Hanlin Ren. Strong average-case lower bounds from non-trivial derandomization. In *Proc. 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1327–1334. ACM, 2020.

[Dam91] Carsten Damm. DET = $\mathrm{L}^{\#\mathsf{L}}$? Informatik-Preprint 8, Fachbereich Informatik der Humboldt-Universität zu Berlin, 1991.

[DGJ^{+}21] Samir Datta, Chetan Gupta, Rahul Jain, Anish Mukherjee, Vimal Raj Sharma, and Raghunath Tewari. Reachability and matching in single crossing minor free graphs. In *Proc. 41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, volume 213 of *LIPIcs*, pages 16:1–16:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[DH10] R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity.* Springer, 2010.

[DMM+16] Arnaud Durand, Meena Mahajan, Guillaume Malod, Nicolas de Rugy-Altherre, and Nitin Saurabh. Homomorphism polynomials complete for VP. *Chic. J. Theor. Comput. Sci.*, 2016, 2016.

[DMR+12] Samir Datta, Meena Mahajan, B. V. Raghavendra Rao, Michael Thomas, and Heribert Vollmer. Counting classes and the fine structure between NC^1 and L. *Theoretical Computer Science*, 417:36–49, 2012.

[EJT10] Michael Elberfeld, Andreas Jakoby, and Till Tantau. Logspace versions of the theorems of Bodlaender and Courcelle. In *Proc. 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152. IEEE Computer Society, 2010.

[Exc12] Stack Exchange. Open problems with monetary rewards. https://mathoverflow.net/questions/66084/open-problems-with-monetary-rewards, 2012.

[Fre12] Rusins Freivalds. Hartmanis-Stearns conjecture on real time and transcendence. In Michael J. Dinneen, Bakhadyr Khoussainov, and André Nies, editors, *Computation, Physics and Beyond - International Workshop on Theoretical Computer Science, WTCS 2012, Dedicated to Cristian S. Calude on the Occasion of His 60th Birthday, Revised Selected and Invited Papers*, volume 7160 of *Lecture Notes in Computer Science*, pages 105–119. Springer, 2012.

[FVB94] Gudmund Skovbjerg Frandsen, Mark Valence, and David A. Mix Barrington. Some results on uniform arithmetic circuit complexity. *Mathematical Systems Theory*, 27(2):105–124, 1994.

[Gil77] John Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977.

[GST19] Chetan Gupta, Vimal Raj Sharma, and Raghunath Tewari. Reachability in O(log n) genus graphs is in unambiguous logspace. In *Proc. 36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *LIPIcs*, pages 34:1–34:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[GSTV14] Brady Garvin, Derrick Stolee, Raghunath Tewari, and N. V. Vinodchandran. ReachFewL = ReachUL. *Computational Complexity*, 23(1):85–98, 2014.

[GW93] Ricard Gavaldà and Osamu Watanabe. On the computational complexity of small descriptions. *SIAM Journal on Computing*, 22(6):1257–1275, 1993.

[GW96] Anna Gál and Avi Wigderson. Boolean complexity classes vs. their arithmetic analogs. *Random Struct. Algorithms*, 9(1-2):99–111, 1996.

[HAB02] William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65(4):695–716, 2002.

[Han06] Kristoffer Arnsfelt Hansen. Constant width planar computation characterizes ACC^0. *Theory of Computing Systems*, 39(1):79–92, 2006.

[Hes01] William Hesse. Division is in uniform TC^0. In *Proc. 28th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2076 of *Lecture Notes in Computer Science*, pages 104–114. Springer, 2001.

[Hir18] Shuichi Hirahara. Non-black-box worst-case to average-case reductions within NP. In *59th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 247–258. IEEE Computer Society, 2018.

[Hir20] Shuichi Hirahara. Unexpected hardness results for Kolmogorov complexity under uniform reductions. In *Proc. 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1038–1051. ACM, 2020.

[HV06] Alexander Healy and Emanuele Viola. Constant-depth circuits for arithmetic in finite fields of characteristic two. In *Proc. 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3884 of *Lecture Notes in Computer Science*, pages 672–683. Springer, 2006.

[Jeř12] Emil Jeřábek. Root finding with threshold circuits. *Theoretical Computer Science*, 462:59–69, 2012.

[Jun85] Hermann Jung. Depth efficient transformations of arithmetic into Boolean circuits. In *Proc. Fundamentals of Computation Theory (FCT)*, volume 199 of *Lecture Notes in Computer Science*, pages 167–174. Springer, 1985.

[Kin12] Jeff Kinne. On TC^0 lower bounds for the permanent. In *Proc. 18th International Conference on Computing and Combinatorics (COCOON)*, volume 7434 of *Lecture Notes in Computer Science*, pages 420–432. Springer, 2012.

[KP09] Pascal Koiran and Sylvain Perifel. A superpolynomial lower bound on the size of uniform non-constant-depth threshold circuits for the permanent. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity (CCC)*, pages 35–40. IEEE Computer Society, 2009.

[Kum96] Martin Kummer. On the complexity of random strings (extended abstract). In *Proc. 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1046 of *Lecture Notes in Computer Science*, pages 25–36. Springer, 1996.

[KV10] Jan Kyncl and Tomás Vyskocil. Logspace reduction of directed reachability for bounded genus graphs to the planar case. *ACM Transactions on Computation Theory*, 1(3):8:1–8:11, 2010.

[Lan97] Klaus-Jörn Lange. An unambiguous class possessing a complete set. In *Proc. 14th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 1200 of *Lecture Notes in Computer Science*, pages 339–350. Springer, 1997.

[LMN10] Nutan Limaye, Meena Mahajan, and Prajakta Nimbhorkar. Longest paths in planar DAGs in unambiguous log-space. *Chicago J. Theor. Comput. Sci.*, 2010, 2010.

[LV19] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 4th Edition.* Texts in Computer Science. Springer, 2019.

[Mah14] Meena Mahajan. Algebraic complexity classes. In *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, volume 26 of *Progress in Computer Science and Applied Logic*, pages 51–75. Springer, 2014.

[Men13] Stefan Mengel. Arithmetic branching programs with memory. In *Proc. 38th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 8087 of *Lecture Notes in Computer Science*, pages 667–678. Springer, 2013.

[MS18] Meena Mahajan and Nitin Saurabh. Some complete and intermediate polynomials in algebraic complexity theory. *Theory of Computing Systems*, 62(3):622–652, 2018.

[MV97] Meena Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chic. J. Theor. Comput. Sci.*, 1997, 1997.

[MVW99] Peter Bro Miltersen, N. V. Vinodchandran, and Osamu Watanabe. Super-polynomial versus half-exponential circuit size in the exponential hierarchy. In *Proc. 5th International Conference on Computing and Combinatorics (COCOON)*, volume 1627 of *Lecture Notes in Computer Science*, pages 210–220. Springer, 1999.

[MW20] Cody D. Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime from a new easy witness lemma. *SIAM Journal on Computing*, 49(5), 2020.

[RA00] Klaus Reinhardt and Eric Allender. Making nondeterminism unambiguous. *SIAM Journal on Computing*, 29:1118–1131, 2000.

[RRW94] Rajesh P. N. Rao, Jörg Rothe, and Osamu Watanabe. Upward separation for FewP and related classes. *Information Processing Letters*, 52(4):175–180, 1994.

[SV85] Sven Skyum and Leslie G. Valiant. A complexity theory based on Boolean algebra. *J. ACM*, 32(2):484–502, 1985.

[Tod91] Seinosuke Toda. Counting problems computationally equivalent to computing the determinant. Technical Report CSIM 91-07, University of Electro-communications, Chofugaoka, May 1991.

[TV07] Luca Trevisan and Salil P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.

[TV12] Raghunath Tewari and N. V. Vinodchandran. Green's theorem and isolation in planar graphs. *Information and Computation*, 215:1–7, 2012.

[TW14] Thomas Thierauf and Fabian Wagner. Reachability in $K_{3,3}$-free and K_5-free graphs is in unambiguous logspace. *Chicago J. Theor. Comput. Sci.*, 2014, 2014.

[Vin91] V. Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *Proc. of the Sixth Annual Structure in Complexity Theory Conference*, pages 270–284. IEEE Computer Society, 1991.

[vMP19] Dieter van Melkebeek and Gautam Prakriya. Derandomizing isolation in space-bounded settings. *SIAM Journal on Computing*, 48(3):979–1021, 2019.

[Vol99] Heribert Vollmer. *Introduction to circuit complexity: a uniform approach.* Springer Science & Business Media, 1999.

[Yap10] Chee Yap. Pi is in log space. Manuscript available at https://cs.nyu.edu/exact/doc/pi-log.pdf, June 2010.