Algebrization: A New Barrier in Complexity Theory*

Scott Aaronson
MIT
aaronson@csail.mit.edu

Avi Wigderson
Institute for Advanced Study
avi@ias.edu

ABSTRACT

Any proof of $\mathsf{P} \neq \mathsf{NP}$ will have to overcome two barriers: *relativization* and *natural proofs*. Yet over the last decade, we have seen circuit lower bounds (for example, that PP does not have linear-size circuits) that overcome both barriers simultaneously. So the question arises of whether there is a third barrier to progress on the central questions in complexity theory.

In this paper we present such a barrier, which we call *algebraic relativization* or *algebrization*. The idea is that, when we relativize some complexity class inclusion, we should give the simulating machine access not only to an oracle A, but also to a low-degree extension of A over a finite field or ring.

We systematically go through basic results and open problems in complexity theory to delineate the power of the new algebrization barrier. First, we show that all known non-relativizing results based on arithmetization—both inclusions such as $\mathsf{IP} = \mathsf{PSPACE}$ and $\mathsf{MIP} = \mathsf{NEXP}$, and separations such as $\mathsf{MA}_{\mathsf{EXP}} \not\subset \mathsf{P/poly}$ —do indeed algebrize. Second, we show that almost all of the major open problems—including P versus NP, P versus RP, and NEXP versus $\mathsf{P/poly}$—will require *non-algebrizing techniques*. In some cases algebrization seems to explain exactly why progress stopped where it did: for example, why we have superlinear circuit lower bounds for $\mathsf{PromiseMA}$ but not for NP.

Our second set of results follows from lower bounds in a new model of *algebraic query complexity*, which we introduce in this paper and which is interesting in its own right. Some of our lower bounds use direct combinatorial and algebraic arguments, while others stem from a surprising connection between our model and communication complexity. Using this connection, we are also able to give an MA-protocol for the Inner Product function with $O\left(\sqrt{n}\log n\right)$ communication (essentially matching a lower bound of Klauck).

*Extended abstract. For the full version, please go to www.scottaaronson.com/papers.

STOC'08, May 17–20, 2008, Victoria, British Columbia, Canada.

Categories and Subject Descriptors

F.1 [**Theory of Computation**]: Computation by Abstract Devices

General Terms

Theory

1. INTRODUCTION

In the history of the P versus NP problem, there were two occasions when researchers stepped back, identified some property of almost all the techniques that had been tried up to that point, and then proved that *no technique with that property could possibly work.* These "meta-discoveries" constitute an important part of what we understand about the P versus NP problem beyond what was understood in 1971.

The first meta-discovery was *relativization.* In 1975, Baker, Gill, and Solovay [4] showed that techniques borrowed from logic and computability theory, such as diagonalization, cannot be powerful enough to resolve P versus NP. For these techniques would work equally well in a "relativized world," where both P and NP machines could compute some function f in a single time step. However, there are some relativized worlds where $\mathsf{P} = \mathsf{NP}$, and other relativized worlds where $\mathsf{P} \neq \mathsf{NP}$. Therefore any solution to the P versus NP problem will require *non-relativizing techniques*: techniques that exploit properties of computation that are specific to the real world.

The second meta-discovery was *natural proofs.* In 1993, Razborov and Rudich [25] analyzed the circuit lower bound techniques that had led to some striking successes in the 1980's, and showed that, if these techniques worked to prove separations like $\mathsf{P} \neq \mathsf{NP}$, then we could turn them around to obtain faster ways to distinguish random functions from pseudorandom functions. But in that case, we would be finding fast algorithms for some of the very same problems (like inverting one-way functions) that we wanted to prove were hard.

1.1 The Need for a New Barrier

Yet for both of these barriers—relativization and natural proofs—we do know ways to circumvent them.

In the early 1990's, researchers managed to prove $\mathsf{IP} = \mathsf{PSPACE}$ [19, 27] and other celebrated theorems about interactive protocols, even in the teeth of relativized worlds where these theorems were false. To do so, they created a new technique called *arithmetization.* The idea was that, instead of treating a Boolean formula φ as just a black box mapping inputs to outputs, one can take advantage of the structure of φ, by "promoting" its AND, OR, or NOT gates to arithmetic operations over some larger field $\mathbb{F}$. One can thereby extend φ to a low-degree polynomial $\widetilde{\varphi} : \mathbb{F}^n \to \mathbb{F}$, which has useful error-correcting properties that were unavailable in the Boolean case.

In the case of the natural proofs barrier, a way to circumvent it was actually known since the work of Hartmanis and Stearns [12] in the 1960's. Any complexity class separation proved via diagonalization—such as $\mathsf{P} \neq \mathsf{EXP}$ or $\Sigma_2^{\mathsf{EXP}} \not\subset \mathsf{P/poly}$ [16]—is inherently non-naturalizing. For diagonalization zeroes in on a specific property of the function f being lower-bounded—namely, the ability of f to simulate a whole class of machines—and thereby avoids the trap of arguing that "f is hard because it looks like a random function."

Until a decade ago, one could at least say that all known circuit lower bounds were subject either to the relativization barrier, *or* to the natural proofs barrier. But not even that is true any more. We now have circuit lower bounds that evade *both* barriers, by cleverly combining arithmetization (which is non-relativizing) with diagonalization (which is non-naturalizing).

The first such lower bound was due to Buhrman, Fortnow, and Thierauf [5], who showed that $\mathsf{MA}_{\mathsf{EXP}}$, the exponential-time analogue of MA, is not in $\mathsf{P/poly}$. To prove that their result was non-relativizing, Buhrman et al. also gave an oracle A such that $\mathsf{MA}_{\mathsf{EXP}}^{A} \subset \mathsf{P}^{A}/\mathsf{poly}$. Using similar ideas, Vinodchandran [29] showed that for every fixed k, the class PP does not have circuits of size n^k; and Aaronson [1] showed that Vinodchandran's result was non-relativizing, by giving an oracle A such that $\mathsf{PP}^A \subset \mathsf{SIZE}^A(n)$. Recently, Santhanam [26] gave a striking improvement of Vinodchandran's result, by showing that for every fixed k, the class $\mathsf{PromiseMA}$ does not have circuits of size n^k.

As Santhanam [26] stressed, these results raise an important question: given that current techniques can already overcome the two central barriers of complexity theory, *how much further can one push those techniques?* Could arithmetization and diagonalization already suffice to prove circuit lower bounds for NEXP, or even $\mathsf{P} \neq \mathsf{NP}$? Or is there a third barrier, beyond relativization and natural proofs, to which even the most recent results are subject?

1.2 Our Contribution

In this paper we show that there is, alas, a third barrier to solving P versus NP and the other central problems of complexity theory.

Recall that a key insight behind the non-relativizing interactive proof results was that, given a Boolean formula φ, one need not treat φ as merely a black box, but can instead reinterpret it as a low-degree polynomial $\widetilde{\varphi}$ over a larger field or ring. To model that insight, in this paper we consider *algebraic oracles*: oracles that can evaluate not only a Boolean function f, but also a low-degree extension $\widetilde{f}$ of f over a finite field or the integers. We then define *algebrization* (short for "algebraic relativization"), the main notion of this paper.

Roughly speaking, we say that a complexity class inclusion $\mathcal{C} \subseteq \mathcal{D}$ *algebrizes* if $\mathcal{C}^A \subseteq \mathcal{D}^{\widetilde{A}}$ for all oracles A and all low-degree extensions $\widetilde{A}$ of A. Likewise, a separation $\mathcal{C} \not\subset \mathcal{D}$ algebrizes if $\mathcal{C}^{\widetilde{A}} \not\subset \mathcal{D}^A$ for all $A, \widetilde{A}$. Notice that algebrization is defined differently for inclusions and separations; and that in both cases, only one complexity class gets the algebraic oracle $\widetilde{A}$, while the other gets the Boolean version A. These subtle asymmetries are essential for this new notion to capture what we want, and will be explained in Section 2.

We demonstrate how algebrization captures a new barrier by proving two sets of results. The first set shows that, of the known results based on arithmetization that fail to relativize, *all* of them algebrize. This includes the interactive proof results, as well as their consequences for circuit lower bounds. More concretely, we show (among other things) that, for all oracles A and low-degree extensions $\widetilde{A}$ of A:

- $\mathsf{PSPACE}^A \subseteq \mathsf{IP}^{\widetilde{A}}$
- $\mathsf{NEXP}^A \subseteq \mathsf{MIP}^{\widetilde{A}}$
- $\mathsf{MA}_{\mathsf{EXP}}^{\widetilde{A}} \not\subset \mathsf{P}^A/\mathsf{poly}$
- $\mathsf{PromiseMA}^{\widetilde{A}} \not\subset \mathsf{SIZE}^A\left(n^k\right)$

The second set of results shows that, for many basic complexity questions, any solution will require non-algebrizing techniques. We show (among other things) that there exist oracles $A, \widetilde{A}$ relative to which:

- $\mathsf{NP}^{\widetilde{A}} \subseteq \mathsf{P}^A$, and indeed $\mathsf{PSPACE}^{\widetilde{A}} \subseteq \mathsf{P}^A$
- $\mathsf{NP}^A \not\subset \mathsf{P}^{\widetilde{A}}$, and indeed $\mathsf{RP}^A \not\subset \mathsf{P}^{\widetilde{A}}$
- $\mathsf{NP}^A \not\subset \mathsf{BPP}^{\widetilde{A}}$, $\mathsf{NP}^A \not\subset \mathsf{BQP}^{\widetilde{A}}$, and $\mathsf{NP}^A \not\subset \mathsf{coMA}^{\widetilde{A}}$
- $\mathsf{NEXP}^{\widetilde{A}} \subset \mathsf{P}^A/\mathsf{poly}$
- $\mathsf{NP}^{\widetilde{A}} \subset \mathsf{SIZE}^A(n)$

These results imply, in particular, that any resolution of the P versus NP problem will need to use non-algebrizing techniques. But the take-home message is stronger: non-algebrizing techniques will be needed even to derandomize RP, to separate NEXP from $\mathsf{P/poly}$, or to prove superlinear circuit lower bounds for NP.

By contrast, recall that the separations $\mathsf{MA}_{\mathsf{EXP}} \not\subset \mathsf{P/poly}$ and $\mathsf{PromiseMA} \not\subset \mathsf{SIZE}\left(n^k\right)$ have already been proved with algebrizing techniques. Thus, we see that known techniques can prove superlinear circuit lower bounds for $\mathsf{PromiseMA}$, but *cannot* do the same for NP—even though $\mathsf{MA} = \mathsf{NP}$ under standard hardness assumptions [18]. Similarly, known techniques can prove superpolynomial circuit lower bounds for $\mathsf{MA}_{\mathsf{EXP}}$ but not for NEXP. To summarize:

> *Algebrization provides nearly the precise limit on*
> *the non-relativizing techniques of the last two decades.*

We speculate that going beyond this limit will require fundamentally new methods.[1]

1.3 Techniques

This section naturally divides into two, one for each of our main sets of results.

[1]While we have shown that most non-relativizing results algebrize, we note that we have skipped some famous examples—involving zero-knowledge protocols for NP, small-depth circuits, time-space tradeoffs for SAT, and the like. We discuss some of these examples in Section 7.

1.3.1 Proving That Existing Results Algebrize

Showing that the interactive proof results algebrize is conceptually simple (though a bit tedious in some cases), once one understands the specific way these results use arithmetization. In our view, it is the very naturalness of the algebrization concept that makes the proofs so simple.

To illustrate, consider the result of Lund, Fortnow, Karloff, and Nisan [19] that $\mathsf{coNP} \subseteq \mathsf{IP}$. In the LFKN protocol, the verifier (Arthur) starts with a Boolean formula φ, which he arithmetizes to produce a low-degree polynomial $\widetilde{\varphi} : \mathbb{F}^n \to \mathbb{F}$. The prover (Merlin) then wants to convince Arthur that $\sum_{x \in \{0,1\}^n} \widetilde{\varphi}(x) = 0$. To do so, Merlin engages Arthur in a conversation about the sums of $\widetilde{\varphi}$ over various subsets of points in $\mathbb{F}^n$. For almost all of this conversation, Merlin is "doing the real work." Indeed, the only time Arthur ever uses his description of $\widetilde{\varphi}$ is in the very last step, when he checks that $\widetilde{\varphi}(r_1, \ldots, r_n)$ is equal to the value claimed by Merlin, for some random field elements $r_1, \ldots, r_n$ chosen earlier in the protocol.

Now suppose we want to prove $\mathsf{coNP}^A \subseteq \mathsf{IP}^{\widetilde{A}}$. The only change is that now Arthur's formula φ will in general contain A gates, in addition to the usual AND, OR, and NOT gates. And therefore, when Arthur arithmetizes φ to produce a low-degree polynomial $\widetilde{\varphi}$, his description of $\widetilde{\varphi}$ will contain terms of the form $A(z_1, \ldots, z_k)$. Arthur then faces the problem of how to evaluate these terms when the inputs $z_1, \ldots, z_k$ are non-Boolean. At this point, though, the solution is clear: Arthur simply calls the oracle $\widetilde{A}$ to get $\widetilde{A}(z_1, \ldots, z_k)$!

While the details are slightly more complicated, the same idea can be used to show $\mathsf{PSPACE}^A \subseteq \mathsf{IP}^{\widetilde{A}}$ and $\mathsf{NEXP}^A \subseteq \mathsf{MIP}^{\widetilde{A}}$.

But what about the non-relativizing separation results, like $\mathsf{MA}_{\mathsf{EXP}}^{\widetilde{A}} \not\subset \mathsf{P}^A/\mathsf{poly}$? When we examine the proofs of these results, we find that each of them combines a single non-relativizing ingredient—namely, an interactive proof result—with a sequence of relativizing results. Therefore, having shown that the interactive proof results algebrize, we have already done most of the work of showing the separations algebrize as well.

1.3.2 Proving The Necessity of Non-Algebrizing Techniques

It is actually easy to show that any proof of $\mathsf{NP} \not\subset \mathsf{P}$ will need non-algebrizing techniques. One simply lets A be a PSPACE-complete language and $\widetilde{A}$ be a PSPACE-complete extension of A; then $\mathsf{NP}^{\widetilde{A}} = \mathsf{P}^A = \mathsf{PSPACE}$. What is harder is to show that any proof of $\mathsf{RP} \subseteq \mathsf{P}$, $\mathsf{NP} \subseteq \mathsf{BPP}$, and so on will need non-algebrizing techniques. For the latter problems, we are faced with the task of proving *algebraic oracle separations*. In other words, we need to show (for example) that there exist oracles $A, \widetilde{A}$ such that $\mathsf{RP}^A \not\subset \mathsf{P}^{\widetilde{A}}$ and $\mathsf{NP}^A \not\subset \mathsf{BPP}^{\widetilde{A}}$.

Just like with standard oracle separations, to prove an algebraic oracle separation one has to do two things:

(1) Prove a concrete lower bound on the query complexity of some function.

(2) Use the query complexity lower bound to diagonalize against a class of Turing machines.

Step (2) is almost the same for algebraic and standard oracle separations; it uses the bounds from (1) in a diagonalization argument. Step (1), on the other hand, is extremely interesting; it requires us to prove lower bounds in a new model of *algebraic query complexity*.

In this model, an algorithm is given oracle access to a Boolean function $A : \{0,1\}^n \to \{0,1\}$. It is trying to answer some question about A—for example, *"is there an* $x \in \{0,1\}^n$ *such that* $A(x) = 1$*?"*—by querying A on various points. The catch is that the algorithm can query not just A itself, but also an adversarially-chosen low-degree extension $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$ of A over some finite field $\mathbb{F}$.[2] In other words, the algorithm is no longer merely searching for a needle in a haystack: it can also search a low-degree extension of the haystack for "nonlocal clues" of the needle's presence!

This model is clearly at least as strong as the standard one, since an algorithm can always restrict itself to Boolean queries only (which are answered identically by A and $\widetilde{A}$). Furthermore, we know from interactive proof results that the new model is sometimes much stronger: sampling points outside the Boolean cube does, indeed, sometimes help a great deal in determining properties of A. This suggests that, to prove lower bounds in this model, we are going to need new techniques.

In this paper we develop *two* techniques for lower-bounding algebraic query complexity, with complementary strengths and weaknesses.

The first technique is based on direct construction of adversarial polynomials. Suppose an algorithm has queried the points $y_1, \ldots, y_t \in \mathbb{F}^n$. Then by a simple linear algebra argument, it is possible to create a multilinear polynomial p that evaluates to 0 on all the y_i's, and that simultaneously has any values we specify on $2^n - t$ points of the Boolean cube. The trouble is that, on the remaining t Boolean points, p will not necessarily be Boolean: that is, p will not necessarily be an extension of a Boolean function. We solve this problem by multiplying p with a *second* multilinear polynomial, to produce a "multiquadratic" polynomial (a polynomial of degree at most 2 in each variable) that is Boolean on the Boolean cube and that also has the desired adversarial behavior.

The idea above becomes more complicated for randomized lower bounds, where we need to argue about the indistinguishability of distributions over multiquadratic polynomials conditioned on a small number of queries. And it becomes more complicated still when (in the full version of this paper) we consider extensions $\widehat{A} : \mathbb{Z}^n \to \mathbb{Z}$ over the integers. In the latter case, we can no longer use linear algebra to construct the multilinear polynomial p, and we need to compensate by bringing in some tools from elementary number theory, namely Chinese remaindering and Hensel lifting. Even then, a technical problem (that the number of bits needed to express $\widehat{A}(x)$ grows with the running times of the machines being diagonalized against) currently prevents us from turning query complexity lower bounds obtained by this technique into algebraic oracle separations over the integers.

Our second lower-bound technique comes as an "unexpected present" from communication complexity. Given a Boolean function $A : \{0,1\}^n \to \{0,1\}$, let A_0 and A_1 be the subfunctions obtained by fixing the first input bit to 0 or 1 respectively. Also, suppose Alice is given the truth table

[2] In the full version of this paper, we also study extensions over the integers.

of A_0, while Bob is given the truth table of A_1. Then we observe the following connection between algebraic query complexity and communication complexity: *If some property of A can be determined using T queries to a multilinear extension $\widetilde{A}$ of A over the finite field $\mathbb{F}$, then it can also be determined by Alice and Bob using $O(Tn \log |\mathbb{F}|)$ bits of communication.*

This connection is extremely generic: it lets us convert randomized algorithms querying $\widetilde{A}$ into randomized communication protocols, quantum algorithms into quantum protocols, MA-algorithms into MA-protocols, and so on. Turning the connection around, we find that *any communication complexity lower bound automatically leads to an algebraic query complexity lower bound.* This means, for example, that we can use celebrated lower bounds for the Disjointness problem [23, 15, 17, 24] to show that there exist oracles $A, \widetilde{A}$ relative to which $\mathsf{NP}^A \not\subset \mathsf{BPP}^{\widetilde{A}}$, and even $\mathsf{NP}^A \not\subset \mathsf{BQP}^{\widetilde{A}}$ and $\mathsf{NP}^A \not\subset \mathsf{coMA}^{\widetilde{A}}$. For the latter two results, we do not know of any proof by direct construction of polynomials.

The communication complexity technique has two further advantages: it yields *multilinear* extensions instead of multiquadratic ones, and it works just as easily over the integers as over finite fields. On the other hand, the lower bounds one gets from communication complexity are more contrived. For example, one can show that solving the Disjointness problem requires exponentially many queries to $\widetilde{A}$, but not that finding a Boolean x such that $A(x) = 1$ does. Also, we do not know how to use communication complexity to construct $A, \widetilde{A}$ such that $\mathsf{NEXP}^{\widetilde{A}} \subset \mathsf{P}^A/\mathsf{poly}$ and $\mathsf{NP}^{\widetilde{A}} \subset \mathsf{SIZE}^A(n)$.

1.4 Related Work

In a survey article on "The Role of Relativization in Complexity Theory," Fortnow [9] defined a class of oracles $\mathcal{O}$ relative to which $\mathsf{IP} = \mathsf{PSPACE}$. His proof that $\mathsf{IP}^A = \mathsf{PSPACE}^A$ for all $A \in \mathcal{O}$ was similar to our proof that $\mathsf{IP} = \mathsf{PSPACE}$ algebrizes. However, because he wanted both complexity classes to have access to the same oracle A, Fortnow had to define his oracles in a subtle recursive way, as follows: start with an arbitrary Boolean oracle B, then let $\widetilde{B}$ be the multilinear extension of B, then let f be the "Booleanization" of $\widetilde{B}$ (i.e., $f(x,i)$ is the i^{th} bit in the binary representation of $\widetilde{B}(x)$), then let $\widetilde{\widetilde{B}}$ be the multilinear extension of f, and so on ad infinitum. Finally let A be the concatenation of all these oracles.

As we discuss in the full version of the paper, it seems extremely difficult to prove *separations* relative to these recursively defined oracles. So if the goal is to show the limitations of current proof techniques for solving open problems in complexity theory, then a non-recursive definition like ours seems essential.

Recently (and independently of us), Juma, Kabanets, Rackoff and Shpilka [14] studied an algebraic query complexity model closely related to ours, and proved lower bounds in this model. In our terminology, they "almost" constructed an oracle A, and a multiquadratic extension $\widetilde{A}$ of A, such that $\#\mathsf{P}^A \not\subset \mathsf{FP}^{\widetilde{A}}/\mathsf{poly}$.[3] Our results in Section 4 extend those of Juma et al. and solve some of their open problems.

Juma et al. also made the interesting observation that, if the extension $\widetilde{A}$ is multilinear rather than multiquadratic, then oracle access to $\widetilde{A}$ sometimes switches from being useless to being extraordinarily powerful. For example, let $A : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$ be the multilinear extension of A, over any field $\mathbb{F}$ of characteristic other than 2. Then we can evaluate the sum $\sum_{x \in \{0,1\}^n} A(x)$ with just a *single* query to $\widetilde{A}$, by using the fact that

$$\sum_{x \in \{0,1\}^n} A(x) = 2^n \widetilde{A}\left(\frac{1}{2}, \ldots, \frac{1}{2}\right).$$

This observation helps to explain why, in Section 4, we will often need to resort to multiquadratic extensions instead of multilinear ones.

2. ORACLES AND ALGEBRIZATION

In this section we discuss some preliminaries, and then formally define the main notions of the paper: extension oracles and algebrization.

Given a multivariate polynomial $p(x_1, \ldots, x_n)$, we define the *multidegree* of p, or $\operatorname{mdeg}(p)$, to be the maximum degree of any x_i. We say p is *multilinear* if $\operatorname{mdeg}(p) \leq 1$, and *multiquadratic* if $\operatorname{mdeg}(p) \leq 2$. Also, we call p an *extension polynomial* if $p(x) \in \{0,1\}$ whenever $x \in \{0,1\}^n$. Intuitively, this means that p is the polynomial extension of some Boolean function $f : \{0,1\}^n \to \{0,1\}$.

The right way to relativize complexity classes such as PSPACE and EXP has long been a subject of dispute: should we allow exponentially-long queries to the oracle, or only polynomially-long queries? On the one hand, if we allow exponentially-long queries, then statements like "$\mathsf{IP} = \mathsf{PSPACE}$ is non-relativizing" are reduced to trivialities, since the PSPACE machine can simply query oracle bits that the IP machine cannot reach. Furthermore the result of Chandra, Kozen, and Stockmeyer [8] that $\mathsf{APSPACE} = \mathsf{EXP}$ becomes non-relativizing, which seems perverse. On the other hand, if we allow only polynomially-long queries, then results based on padding—for example, $\mathsf{P} = \mathsf{NP} \Longrightarrow \mathsf{EXP} = \mathsf{NEXP}$—will generally fail to relativize.[4]

In this paper we adopt a pragmatic approach, writing $\mathcal{C}^A$ or $\mathcal{C}^{A[\mathrm{poly}]}$ to identify which convention we have in mind. More formally:

Definition 2.1 (Oracle). *An oracle A is a collection of Boolean functions $A_m : \{0,1\}^m \to \{0,1\}$, one for each $m \in \mathbb{N}$. Then given a complexity class $\mathcal{C}$, by $\mathcal{C}^A$ we mean the class of languages decidable by a $\mathcal{C}$ machine that can query A_m for any m of its choice. By $\mathcal{C}^{A[\mathrm{poly}]}$ we mean the class of languages decidable by a $\mathcal{C}$ machine that, on inputs of length n, can query A_m for any $m = O(\mathrm{poly}(n))$. For classes $\mathcal{C}$ such that all computation paths are polynomially bounded (for example, P, NP, BPP, $\#\mathsf{P}$...), it is obvious that $\mathcal{C}^{A[\mathrm{poly}]} = \mathcal{C}^A$.*

We now define the key notion of an *extension oracle* over a finite field.

[3] We say "almost" because they did not ensure $\widetilde{A}(x)$ was Boolean for all Boolean x; this is an open problem of theirs that we solve in Section 4.2.1. Also, their result is only for field extensions and not integer extensions.

[4] Indeed, let A be any PSPACE-complete language. Then $\mathsf{P}^A = \mathsf{NP}^A$, but $\mathsf{EXP}^{A[\mathrm{poly}]} = \mathsf{NEXP}^{A[\mathrm{poly}]}$ if and only if $\mathsf{EXP} = \mathsf{NEXP}$ in the unrelativized world.

Definition 2.2 (Extension). *Let $A_m : \{0,1\}^m \to \{0,1\}$ be a Boolean function, and let $\mathbb{F}$ be a finite field. Then an extension of A_m over $\mathbb{F}$ is a polynomial $\widetilde{A}_{m,\mathbb{F}} : \mathbb{F}^m \to \mathbb{F}$ such that $\widetilde{A}_{m,\mathbb{F}}(x) = A_m(x)$ whenever $x \in \{0,1\}^m$. Also, given an oracle $A = (A_m)$, an extension $\widetilde{A}$ of A is a collection of polynomials $\widetilde{A}_{m,\mathbb{F}} : \mathbb{F}^m \to \mathbb{F}$, one for each positive integer m and finite field $\mathbb{F}$, such that*

(i) $\widetilde{A}_{m,\mathbb{F}}$ is an extension of A_m for all $m, \mathbb{F}$, and

(ii) there exists a constant c such that $\operatorname{mdeg}(\widetilde{A}_{m,\mathbb{F}}) \le c$ for all $m, \mathbb{F}$.[5]

Then given a complexity class $\mathcal{C}$, by $\mathcal{C}^{\widetilde{A}}$ we mean the class of languages decidable by a $\mathcal{C}$ machine that can query $\widetilde{A}_{m,\mathbb{F}}$ for any integer m and finite field $\mathbb{F}$. By $\mathcal{C}^{\widetilde{A}[\mathrm{poly}]}$ we mean the class of languages decidable by a $\mathcal{C}$ machine that, on inputs of length n, can query $\widetilde{A}_{m,\mathbb{F}}$ for any integer $m = O(\mathrm{poly}(n))$ and finite field with $|\mathbb{F}| = 2^{O(m)}$.

We use $\operatorname{mdeg}(\widetilde{A})$ to denote the maximum multidegree of any $\widetilde{A}_m$.

In this extended abstract, we will restrict ourselves to extensions over *finite fields*, as they are easier to work with than integer extensions and let us draw almost the same conceptual conclusions. We note that many of our results (including all results showing that existing results algebrize, and all oracle separations proved via communication complexity) easily carry over to the integer setting. Furthermore, even our oracle separations proved via direct construction can be "partly" carried over to the integer setting. The full version of the paper studies integer extensions in more detail.

Definition 2.3 (Algebrization). *We say the complexity class inclusion $\mathcal{C} \subseteq \mathcal{D}$ algebrizes if $\mathcal{C}^A \subseteq \mathcal{D}^{\widetilde{A}}$ for all oracles A and all finite field extensions $\widetilde{A}$ of A. Likewise, we say that $\mathcal{C} \subseteq \mathcal{D}$ does not algebrize, or that proving $\mathcal{C} \subseteq \mathcal{D}$ would require non-algebrizing techniques, if there exist $A, \widetilde{A}$ such that $\mathcal{C}^A \not\subset \mathcal{D}^{\widetilde{A}}$.*

We say the separation $\mathcal{C} \not\subset \mathcal{D}$ algebrizes if $\mathcal{C}^{\widetilde{A}} \not\subset \mathcal{D}^A$ for all $A, \widetilde{A}$. Likewise, we say that $\mathcal{C} \not\subset \mathcal{D}$ does not algebrize, or that proving $\mathcal{C} \not\subset \mathcal{D}$ would require non-algebrizing techniques, if there exist $A, \widetilde{A}$ such that $\mathcal{C}^{\widetilde{A}} \subseteq \mathcal{D}^A$.

When we examine the above definition, two questions arise. First, why can one complexity class access the extension $\widetilde{A}$, while the other class can only access the Boolean part A? And second, why is it the "right-hand class" that can access $\widetilde{A}$ for inclusions, but the "left-hand class" that can access $\widetilde{A}$ for separations?

The basic answer is that, under a more stringent notion of algebrization, we would not know how to prove that existing interactive proof results algebrize. So for example, while we will prove that $\mathsf{PSPACE}^{A[\mathrm{poly}]} \subseteq \mathsf{IP}^{\widetilde{A}}$ for all oracles A and extensions $\widetilde{A}$ of A, we do not know how to prove that $\mathsf{PSPACE}^{\widetilde{A}[\mathrm{poly}]} = \mathsf{IP}^{\widetilde{A}}$ for all $\widetilde{A}$.

[5]All of our results would work equally well if we instead chose to limit $\operatorname{mdeg}(\widetilde{A}_{m,\mathbb{F}})$ by a linear or polynomial function of m. On the other hand, nowhere in this paper will $\operatorname{mdeg}(\widetilde{A}_{m,\mathbb{F}})$ need to be greater than 2.

3. WHY EXISTING TECHNIQUES ALGEBRIZE

In the full version of the paper, we go through existing non-relativizing results based on arithmetization, and show that they can *all* be recast in algebrizing form. The details are mostly omitted here due to space limitations.

As one prototypical example, though, let us sketch why the result of Lund, Fortnow, Karloff, and Nisan [19] that $\#\mathsf{P} \subset \mathsf{FP/poly} \Rightarrow \mathsf{P}^{\#\mathsf{P}} = \mathsf{MA}$ is algebrizing.

Theorem 3.1. *For all $A, \widetilde{A}$, if $\#\mathsf{P}^{\widetilde{A}} \subset \mathsf{FP}^{\widetilde{A}}/\mathsf{poly}$ then $\mathsf{P}^{\#\mathsf{P}^A} \subseteq \mathsf{MA}^{\widetilde{A}}$.*

Proof Sketch. Let $\#SAT^A$ be the $\#\mathsf{P}^A$-complete problem in which we are given a Boolean formula F^A consisting of AND, OR, NOT, and A-oracle gates, and want to count the number of inputs that cause F^A to accept. Then it suffices to show how to solve $\#SAT^A$ problems in $\mathsf{MA}^{\widetilde{A}}$, assuming $\#\mathsf{P}^{\widetilde{A}} \subset \mathsf{FP}^{\widetilde{A}}/\mathsf{poly}$. The procedure is simply this: first guess a $\mathsf{FP}^{\widetilde{A}}/\mathsf{poly}$ circuit for $\#\mathsf{P}^{\widetilde{A}}$; then use that circuit to simulate the prover in an interactive protocol for $\#SAT^A$. The interactive protocol in question is just the usual one due to Lund et al. [19]—with the one further detail that, when arithmetizing the Boolean formula F^A, Arthur replaces every A-gate by an $\widetilde{A}$-gate, which he then calls the extension oracle $\widetilde{A}$ to evaluate.[6] $\square$

Using similar ideas, one can show that the famous results $\mathsf{IP} = \mathsf{PSPACE}$ and $\mathsf{MIP} = \mathsf{NEXP}$, due to Shamir [27] and Babai, Fortnow, and Lund [3] respectively, are also algebrizing:

Theorem 3.2. *For all $A, \widetilde{A}$, $\mathsf{PSPACE}^{A[\mathrm{poly}]} \subseteq \mathsf{IP}^{\widetilde{A}}$.*

Theorem 3.3. *For all $A, \widetilde{A}$, $\mathsf{NEXP}^{A[\mathrm{poly}]} \subseteq \mathsf{MIP}^{\widetilde{A}}$.*

Let $\mathsf{MIP}_{\mathsf{EXP}}$ be the subclass of MIP where the provers are in EXP. Then Babai, Fortnow, and Lund [3] showed that $\mathsf{MIP}_{\mathsf{EXP}} = \mathsf{EXP}$. We can likewise show the following:

Theorem 3.4. *For all $A, \widetilde{A}$, $\mathsf{EXP}^{A[\mathrm{poly}]} \subseteq \mathsf{MIP}^{\widetilde{A}}_{\mathsf{EXP}}$.*

Finally let us consider non-relativizing circuit lower bounds, such as $\mathsf{PP} \not\subset \mathsf{SIZE}(n^k)$ and $\mathsf{MA}_{\mathsf{EXP}} \not\subset \mathsf{P/poly}$. The key point is that each of these results actually has a conditional collapse as its only non-relativizing ingredient. So having shown that the conditional collapses algebrize, we have already done most of the work of showing that the circuit lower bounds algebrize as well.

To illustrate, let us now use Theorem 3.1 to show that the $\mathsf{MA}_{\mathsf{EXP}} \not\subset \mathsf{P/poly}$ theorem of Buhrman, Fortnow, and Thierauf [5] algebrizes.

Theorem 3.5. *For all $A, \widetilde{A}$, we have $\mathsf{MA}^{\widetilde{A}}_{\mathsf{EXP}} \not\subset \mathsf{P}^A/\mathsf{poly}$.*

[6]Unlike for $\#SAT^A$, we do *not* know how to give an $\mathsf{IP}^{\widetilde{A}}$ protocol for $\#SAT^{\widetilde{A}}$—intuitively because a $\#SAT^{\widetilde{A}}$ formula could query $\widetilde{A}$ in ways that do not respect $\widetilde{A}$'s structure as a polynomial. This is why, for example, we can only show that $\mathsf{P}^{\#\mathsf{P}^A} \subseteq \mathsf{IP}^{\widetilde{A}}$ and not that $\mathsf{P}^{\#\mathsf{P}^{\widetilde{A}}} \subseteq \mathsf{IP}^{\widetilde{A}}$.

PROOF. Suppose $\mathsf{MA}_{\mathsf{EXP}}^{\widetilde{A}} \subset \mathsf{P}^{A}/\mathsf{poly} \subseteq \mathsf{P}^{\widetilde{A}}/\mathsf{poly}$. Then certainly $\mathsf{P}^{\#\mathsf{P}^{\widetilde{A}}} \subset \mathsf{P}^{\widetilde{A}}/\mathsf{poly}$ as well, so Theorem 3.1 implies that $\mathsf{P}^{\#\mathsf{P}^{A}} \subseteq \mathsf{MA}^{\widetilde{A}}$. Hence we also have $\left(\Sigma_2^{\mathsf{P}}\right)^A \subseteq \mathsf{MA}^{\widetilde{A}}$ by Toda's Theorem [28], and hence $\left(\Sigma_2^{\mathsf{EXP}}\right)^A \subseteq \mathsf{MA}_{\mathsf{EXP}}^{\widetilde{A}}$ by padding. But Kannan's Theorem [16] tells us that $\left(\Sigma_2^{\mathsf{EXP}}\right)^A \not\subset \mathsf{P}^A/\mathsf{poly}$, so $\mathsf{MA}_{\mathsf{EXP}}^{\widetilde{A}} \not\subset \mathsf{P}^A/\mathsf{poly}$ as well. $\square$

Using a tighter version of Theorem 3.1, we can also show that the recent $\mathsf{PromiseMA} \not\subset \mathsf{SIZE}\left(n^k\right)$ theorem of Santhanam [26] algebrizes:

THEOREM 3.6. *For all $A, \widetilde{A}$ and constants k, we have* $\mathsf{PromiseMA}^{\widetilde{A}} \not\subset \mathsf{SIZE}^A\left(n^k\right)$.

4. ALGEBRAIC QUERY COMPLEXITY

What underlies our algebraic oracle separations is a new model of *algebraic query complexity*. In the standard query complexity model, an algorithm is trying to compute some property of a Boolean function $A : \{0,1\}^n \to \{0,1\}$ by querying A on various points. In our model, the function $A : \{0,1\}^n \to \{0,1\}$ is still Boolean, but the algorithm is allowed to query not just A, but also a low-degree extension $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$ of A over some field $\mathbb{F}$. In this section we develop the algebraic query complexity model in its own right, and prove several lower bounds in this model. Then, in Section 5, we apply our lower bounds to prove algebraic oracle separations. The full version of the paper considers the variant where the algorithm can query an extension of A over the integers.

Throughout this section we let $N = 2^n$. Algorithms will compute Boolean functions (properties) $f : \{0,1\}^N \to \{0,1\}$. An input A to f will be viewed interchangeably as an N-bit string $A \in \{0,1\}^N$, or as a Boolean function $A : \{0,1\}^n \to \{0,1\}$ of which the string is the truth table.

Let us recall some standard query complexity measures. Given a Boolean function $f : \{0,1\}^N \to \{0,1\}$, the *deterministic query complexity* of f, or $\mathrm{D}(f)$, is defined to be the minimum number of queries made by any deterministic algorithm that evaluates f on every input. Likewise, the (bounded-error) *randomized query complexity* $\mathrm{R}(f)$ is defined to be the minimum expected[7] number of queries made by any randomized algorithm that evaluates f with probability at least $2/3$ on every input. The bounded-error *quantum query complexity* $\mathrm{Q}(f)$ is defined analogously, with quantum algorithms in place of randomized ones. See Buhrman and de Wolf [7] for a survey of these measures.

We now define similar measures for algebraic query complexity. In our definition, an important parameter will be the multidegree of the allowed extension (recall that $\mathrm{mdeg}(p)$ is the largest degree of any of the variables of p). In all of our results, this parameter will be either 1 or 2.

DEFINITION 4.1. *Let $f : \{0,1\}^N \to \{0,1\}$ be a Boolean function, let $\mathbb{F}$ be any field, and let c be a positive integer. Also, let $\mathcal{M}$ be the set of deterministic algorithms M such that $M^{\widetilde{A}}$ outputs $f(A)$ for every oracle $A : \{0,1\}^n \to \{0,1\}$ and every finite field extension $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$ of A with $\mathrm{mdeg}(\widetilde{A}) \leq c$. Then the deterministic algebraic query complexity of f over $\mathbb{F}$ is defined as*

$$\widetilde{\mathrm{D}}_{\mathbb{F},c}(f) := \min_{M \in \mathcal{M}} \max_{A,\widetilde{A}\ :\ \mathrm{mdeg}(\widetilde{A}) \leq c} T_M(\widetilde{A}),$$

where $T_M(\widetilde{A})$ is the number of queries to $\widetilde{A}$ made by $M^{\widetilde{A}}$. The randomized and quantum algebraic query complexities $\widetilde{\mathrm{R}}_{\mathbb{F},c}(f)$ and $\widetilde{\mathrm{Q}}_{\mathbb{F},c}(f)$ are defined similarly, except with bounded-error randomized and quantum algorithms in place of deterministic ones.

[7] Or the worst-case number of queries: up to the exact constant in the success probability, one can always ensure that this is about the same as the expected number.

4.1 Multilinear Polynomials

The construction of "adversary polynomials" in our lower bound proofs will require some useful facts about multilinear polynomials. In particular, the basis of delta functions for these polynomials will come in handy.

In what follows $\mathbb{F}$ is an arbitrary field (finite or infinite). Given a Boolean point z, define

$$\delta_z(x) := \prod_{i : z_i = 1} x_i \prod_{i : z_i = 0} (1 - x_i)$$

to be the unique multilinear polynomial that is 1 at z and 0 elsewhere on the Boolean cube. Then for an arbitrary multilinear polynomial $m : \mathbb{F}^n \to \mathbb{F}$, we can write m uniquely in the basis of δ_z's as follows:

$$m(x) = \sum_{z \in \{0,1\}^n} m_z \delta_z(x)$$

We will often identify a multilinear polynomial m with its coefficients m_z in this basis. Note that for any Boolean point z, the value $m(z)$ is simply the coefficient m_z in the above representation.

4.2 Lower Bounds by Direct Construction

We now prove lower bounds on algebraic query complexity over fields. The goal will be to show that querying points outside the Boolean cube is useless if one wants to gain information about values on the Boolean cube. In full generality, this is of course false (as witnessed by interactive proofs and PCPs on the one hand, and by the result of Juma et al. [14] on the other). To make our adversary arguments work, it will be crucial to give ourselves sufficient freedom, by using polynomials of multidegree 2 rather than multilinear polynomials.

We first prove deterministic lower bounds, which are quite simple, and then extend them to probabilistic lower bounds. Both work for the natural NP predicate of finding a Boolean point z such that $A(z) = 1$.

4.2.1 Deterministic Lower Bounds

LEMMA 4.2. *Let $\mathbb{F}$ be a field and let $y_1, \ldots, y_t$ be points in $\mathbb{F}^n$. Then there exists a multilinear polynomial $m : \mathbb{F}^n \to \mathbb{F}$ such that*

(i) $m(y_i) = 0$ for all $i \in [t]$, and
(ii) $m(z) = 1$ for at least $2^n - t$ Boolean points z.

PROOF. If we represent m as

$$m(x) = \sum_{z \in \{0,1\}^n} m_z \delta_z(x),$$

then the constraint $m(y_i) = 0$ for all $i \in [t]$ corresponds to t linear equations over $\mathbb{F}$ relating the 2^n coefficients m_z. By basic linear algebra, it follows that there must be a solution in which at least $2^n - t$ of the m_z's are set to 1, and hence $m(z) = 1$ for at least $2^n - t$ Boolean points z. □

Lemma 4.3. *Let $\mathbb{F}$ be a field and let $y_1, \ldots, y_t$ be points in $\mathbb{F}^n$. Then for at least $2^n - t$ Boolean points $w \in \{0,1\}^n$, there exists a multiquadratic extension polynomial $p : \mathbb{F}^n \to \mathbb{F}$ such that*

(i) $p(y_i) = 0$ for all $i \in [t]$,
(ii) $p(w) = 1$, and
(iii) $p(z) = 0$ for all Boolean $z \neq w$.

Proof. Let $m : \mathbb{F}^n \to \mathbb{F}$ be the multilinear polynomial from Lemma 4.2, and pick any Boolean w such that $m(w) = 1$. Then a multiquadratic extension polynomial p satisfying properties (i)-(iii) can be obtained from m as follows:

$$p(x) := m(x)\,\delta_w(x).$$

□

Given a Boolean function $A : \{0,1\}^n \to \{0,1\}$, let the OR problem be that of deciding whether there exists an $x \in \{0,1\}^n$ such that $A(x) = 1$. Then Lemma 4.3 easily yields an exponential lower bound on the algebraic query complexity of the OR problem.

Theorem 4.4. $\widetilde{\mathrm{D}}_{\mathbb{F},2}(\mathrm{OR}) = 2^n$ *for every field $\mathbb{F}$.*

Proof. Let $\mathcal{Y}$ be the set of points queried by a deterministic algorithm, and suppose $|\mathcal{Y}| < 2^n$. Then Lemma 4.3 implies that there exists a multiquadratic extension polynomial $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$ such that $\widetilde{A}(y) = 0$ for all $y \in \mathcal{Y}$, but $\widetilde{A}(w) = 1$ for some Boolean w. So even if the algorithm is adaptive, we can let $\mathcal{Y}$ be the set of points it queries *assuming each query is answered with* 0, and then find $\widetilde{A}, \widetilde{B}$ such that $\widetilde{A}(y) = \widetilde{B}(y) = 0$ for all $y \in \mathcal{Y}$, but nevertheless $\widetilde{A}$ and $\widetilde{B}$ lead to different values of the OR function. □

Again, the results of Juma et al. [14] imply that multidegree 2 is essential here, since for multilinear polynomials it is possible to solve the OR problem with only *one* query (over fields of characteristic greater than 2).

4.2.2 Probabilistic Lower Bounds

In the full version of this paper, we generalize Theorem 4.4 to a lower bound against randomized algorithms. As usual, this is done via the Yao minimax principle, namely by constructing a distribution over oracles which is hard for every deterministic algorithm that queries few points.

Lemma 4.5. *Let $\mathbb{F}$ be a finite field.*[8] *Also, for all $w \in \{0,1\}^n$, let D_w be the uniform distribution over multiquadratic polynomials $p : \mathbb{F}^n \to \mathbb{F}$ such that $p(w) = 1$ and $p(z) = 0$ for all Boolean $z \neq w$. Suppose an adversary chooses a "marked point" $w \in \{0,1\}^n$ uniformly at random, and then chooses p according to D_w. Then any deterministic algorithm, after making t queries to p, will have queried w with probability at most $t/2^n$.*

An immediate corollary of Lemma 4.5 is that, over a finite field, randomized algebraic query algorithms do no better than deterministic ones at evaluating the OR function.

Theorem 4.6. $\widetilde{\mathrm{R}}_{\mathbb{F},2}(\mathrm{OR}) = \Omega(2^n)$ *for every finite field $\mathbb{F}$.*

[8]Note that we are only able to prove this lemma for *finite* fields—the reason being that we need to consider a *uniform* distribution over all polynomials with given restrictions.

4.3 Lower Bounds by Communication Complexity

In this section we point out a simple connection between algebraic query complexity and communication complexity. Specifically, we show that *algebraic* query algorithms can be efficiently simulated by *Boolean* communication protocols. This connection allows us to derive many lower bounds on algebraic query complexity that we do not know how to prove with the direct techniques of the previous section.

For concreteness, we first state our "transfer principle" for deterministic query and communication complexities—but as we will see, the principle is much broader.

Theorem 4.7. *Let $A : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let $\widetilde{A} : \mathbb{F}_q^n \to \mathbb{F}_q$ be the unique multilinear extension of A over a finite field $\mathbb{F}$. Suppose one can evaluate some Boolean predicate f of A using T deterministic adaptive queries to $\widetilde{A}$. Also, let A_0 and A_1 be the subfunctions of A obtained by restricting the first bit to 0 or 1 respectively. Then if Alice is given the truth table of A_0 and Bob is given the truth table of A_1, they can jointly evaluate $f(A)$ using $O(Tn \log|\mathbb{F}|)$ bits of communication.*

Proof. Given any point $y \in \mathbb{F}^n$, we can write $\widetilde{A}(y)$ as a linear combination of the values taken by A on the Boolean cube, like so:

$$\widetilde{A}(y) = \sum_{z \in \{0,1\}^n} \delta_z(y)\,A(z).$$

Now let M be an algorithm that evaluates f using T queries to $\widetilde{A}$. Our communication protocol will simply perform a step-by-step simulation of M, as follows. Let $y_1 \in \mathbb{F}^n$ be the first point queried by M. Then Alice computes the partial sum

$$\widetilde{A}_0(y_1) := \sum_{z \in \{0,1\}^{n-1}} \delta_{0z}(y)\,A(0z)$$

and sends $(y_1, \widetilde{A}_0(y_1))$ to Bob. Next Bob computes

$$\widetilde{A}_1(y_1) := \sum_{z \in \{0,1\}^{n-1}} \delta_{1z}(y)\,A(1z),$$

from which he learns $\widetilde{A}(y_1) = \widetilde{A}_0(y_1) + \widetilde{A}_1(y_1)$. Bob can then determine y_2, the second point queried by M given that the first query had outcome $\widetilde{A}(y_1)$. So next Bob computes $\widetilde{A}_1(y_2)$ and sends $(y_2, \widetilde{A}_1(y_2))$ to Alice. Next Alice computes $\widetilde{A}(y_2) = \widetilde{A}_0(y_2) + \widetilde{A}_1(y_2)$, determines y_3, and sends $(y_3, \widetilde{A}_0(y_3))$ to Bob, and so on for T rounds. Each message uses $O(n \log|\mathbb{F}|)$ bits, from which it follows that the total communication cost is $O(Tn \log|\mathbb{F}|)$. □

In proving Theorem 4.7, notice that we never needed the assumption that M was deterministic. Had M been randomized, our simulation would have produced a randomized protocol; had M been quantum, it would have produced a quantum protocol; had M been an MA machine, it would have produced an MA protocol, and so on.

To illustrate the power of Theorem 4.7, let us now prove a lower bound on algebraic query complexity without using anything about polynomials.

Given two Boolean strings $x = x_1 \ldots x_N$ and $y = y_1 \ldots y_N$, recall that the Disjointness problem is to decide whether there exists an index $i \in [N]$ such that $x_i = y_i = 1$. Supposing that Alice holds x and Bob holds y, Kalyasundaram and Schnitger [15] showed that any randomized protocol to solve this problem requires Alice and Bob to exchange $\Omega(N)$ bits (see also the simpler proof by Razborov [23]).

In our setting, the problem becomes the following: given a Boolean function $A : \{0,1\}^n \to \{0,1\}$, decide whether there exists an $x \in \{0,1\}^{n-1}$ such that $A(0x) = A(1x) = 1$. Call this problem DISJ, and suppose we want to solve DISJ using a randomized algorithm that queries the multilinear extension $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$ of A. Then Theorem 4.7 immediately yields a lower bound on the number of queries we need:

THEOREM 4.8. $\widetilde{\mathrm{R}}_{\mathbb{F},1}(\mathrm{DISJ}) = \Omega\left(\frac{2^n}{n \log|\mathbb{F}|}\right)$ *for all finite fields* $\mathbb{F}$.

PROOF. Suppose $\widetilde{\mathrm{R}}_{\mathbb{F},1}(\mathrm{DISJ}) = o\left(\frac{2^n}{n \log|\mathbb{F}|}\right)$. Then by Theorem 4.7, we get a randomized protocol for the Disjointness problem with communication cost $o(N)$, where $N = 2^{n-1}$. But this contradicts [23, 15]. $\square$

5. THE NEED FOR NON-ALGEBRIZING TECHNIQUES

In this section we show that solving many of the open problems in complexity theory will require non-algebrizing techniques. We have already done much of the work in Section 4, by proving lower bounds on algebraic query complexity. What remains is to combine these query complexity results with diagonalization-type arguments, in order to achieve the oracle separations and collapses we want.

5.1 Non-Algebrizing Techniques Needed for P vs. NP

We start with an easy but fundamental result: that *any proof of* $\mathsf{P} \neq \mathsf{NP}$ *will require non-algebrizing techniques.*

THEOREM 5.1. *There exist* $A, \widetilde{A}$ *such that* $\mathsf{NP}^{\widetilde{A}} \subseteq \mathsf{P}^{A}$.

PROOF. Let A be any PSPACE-complete language, and let $\widetilde{A}$ be the unique multilinear extension of A. As observed by Babai, Fortnow, and Lund [3], the multilinear extension of any PSPACE language is also in PSPACE. So as in the usual argument of Baker, Gill, and Solovay [4], we have $\mathsf{NP}^{\widetilde{A}} = \mathsf{NP}^{\mathsf{PSPACE}} = \mathsf{PSPACE} = \mathsf{P}^{A}$. $\square$

Note that the same argument immediately implies that any proof of $\mathsf{P} \neq \mathsf{PSPACE}$ will require non-algebrizing techniques.

Next we show that any proof of $\mathsf{P} = \mathsf{NP}$ would require non-algebrizing techniques, by giving an algebraic oracle *separation* between P and NP.

THEOREM 5.2. *There exist* $A, \widetilde{A}$ *such that* $\mathsf{NP}^{A} \not\subset \mathsf{P}^{\widetilde{A}}$. *Furthermore, the language* L *that achieves the separation simply corresponds to deciding, on inputs of length* n*, whether there exists a* $w \in \{0,1\}^n$ *with* $A_n(w) = 1$.

PROOF SKETCH. The proof closely follows the usual diagonalization argument of Baker, Gill, and Solovay [4]. The only difference is that we have to use a variant of Lemma 4.3 to handle the fact that the P machine can query a low-degree extension. More precisely, for every n, the oracle A contains a Boolean function $A_n : \{0,1\}^n \to \{0,1\}$, while $\widetilde{A}$ contains a multiquadratic extension $\widetilde{A}_{n,\mathbb{F}} : \mathbb{F}^n \to \mathbb{F}$ of A_n for every n and finite field $\mathbb{F}$. Let L be the unary language consisting of all strings 1^n for which there exists a $w \in \{0,1\}^n$ such that $A_n(w) = 1$. Then clearly $L \in \mathsf{NP}^{A}$ for all A. On the other hand, by using a variant of Lemma 4.3, it is not hard to choose $A, \widetilde{A}$ so that $L \notin \mathsf{P}^{\widetilde{A}}$. (Details are deferred to the full version.) $\square$

The same idea also yields the stronger result that there exist $A, \widetilde{A}$ such that $\mathsf{RP}^{A} \not\subset \mathsf{P}^{\widetilde{A}}$. Indeed, by interleaving oracles such that $\mathsf{RP}^{A} \not\subset \mathsf{P}^{\widetilde{A}}$ and $\mathsf{coRP}^{A} \not\subset \mathsf{P}^{\widetilde{A}}$, it is also possible to construct $A, \widetilde{A}$ such that $\mathsf{ZPP}^{A} \not\subset \mathsf{P}^{\widetilde{A}}$.

5.2 Non-Algebrizing Techniques Needed for Circuit Lower Bounds

In the full version of this paper, we combine (i) a variant of Lemma 4.3 with (ii) a standard forcing proof that there exists an oracle A such that $\mathsf{NTIME}^{A}(2^n) \subset \mathsf{SIZE}^{A}(n)$, to obtain the following:

THEOREM 5.3. *There exist oracles* $A, \widetilde{A}$ *such that* $\mathsf{NTIME}^{\widetilde{A}}(2^n) \subset \mathsf{SIZE}^{A}(n)$.

By a padding argument, Theorem 5.3 immediately gives $A, \widetilde{A}$ such that $\mathsf{NEXP}^{\widetilde{A}} \subset \mathsf{P}^{A}/\mathsf{poly}$. This then implies that any proof of $\mathsf{NEXP} \not\subset \mathsf{P}/\mathsf{poly}$ will require non-algebrizing techniques. Note that this is almost the best result possible, since Theorem 3.5 implies that there do *not* exist $A, \widetilde{A}$ such that $\mathsf{MA}_{\mathsf{EXP}}^{\widetilde{A}} \subset \mathsf{P}^{A}/\mathsf{poly}$.

Wilson [30] gave an oracle A relative to which $\mathsf{EXP}^{\mathsf{NP}^{A}} \subset \mathsf{P}^{A}/\mathsf{poly}$. Using similar ideas, one can generalize the construction of Theorem 5.3 to obtain $A, \widetilde{A}$ such that $\mathsf{EXP}^{\mathsf{NP}^{\widetilde{A}}} \subset \mathsf{P}^{A}/\mathsf{poly}$. One can also give $A, \widetilde{A}$ such that $\mathsf{BPEXP}^{\widetilde{A}} \subset \mathsf{P}^{A}/\mathsf{poly}$. We omit the details.

5.3 Non-Algebrizing Techniques Needed for Other Problems

Using the communication complexity transfer principle from Section 4.3, we can convert essentially *any* separation of communication complexity classes into the corresponding separation in the algebraic oracle world. So for example, by using communication complexity lower bounds due to Kalyasundaram and Schnitger [15], Klauck [17], Razborov [24], Raz [20], Raz and Shpilka [21], and Buhrman et al. [6] respectively, we are able to show the following.

THEOREM 5.4. *There exist* $A, \widetilde{A}$ *such that*

(i) $\mathsf{NP}^{A} \not\subset \mathsf{BPP}^{\widetilde{A}}$,

(ii) $\mathsf{coNP}^{A} \not\subset \mathsf{MA}^{\widetilde{A}}$,

(iii) $\mathsf{NP}^{A} \not\subset \mathsf{BQP}^{\widetilde{A}}$,

(iv) $\mathsf{BQP}^{A} \not\subset \mathsf{BPP}^{\widetilde{A}}$,

(v) $\mathsf{QMA}^{A} \not\subset \mathsf{MA}^{\widetilde{A}}$, *and*

(vi) $\mathsf{P}^{\mathsf{NP}^{A}} \not\subset \mathsf{PP}^{\widetilde{A}}$.

Furthermore, for all of these separations $\widetilde{A}$ *is simply the multilinear extension of* A.

6. APPLICATION TO COMMUNICATION COMPLEXITY

Klauck [17] showed that any MA-protocol for the Disjointness problem has communication cost $\Omega(\sqrt{n})$. The "natural" conjecture would be that the $\sqrt{n}$ was merely an artifact of his proof, and that a more refined argument would yield the optimal lower bound of $\Omega(n)$. However, using a protocol directly inspired by our algebrization framework, we are able to show that this conjecture is false. Below, we give an $O(\sqrt{n}\log n)$-communication MA-protocol (which is nearly optimal) not only for Disjointness, but also for the Inner Product problem, where Alice and Bob want to compute $IP(x,y) := \sum_{i=1}^{n} x_i y_i$ as an integer.

THEOREM 6.1. *There exist* MA*-protocols for the Disjointness and Inner Product problems, in which Alice receives an $O(\sqrt{n}\log n)$-bit witness from Merlin and an $O(\sqrt{n}\log n)$-bit message from Bob.*

PROOF. It suffices to give a protocol for Inner Product; a protocol for Disjointness then follows immediately. Assume n is a perfect square. Then Alice and Bob can be thought of as holding functions $a : [\sqrt{n}] \times [\sqrt{n}] \to \{0,1\}$ and $b : [\sqrt{n}] \times [\sqrt{n}] \to \{0,1\}$ respectively. Their goal is to compute the inner product

$$IP := \sum_{x,y \in [\sqrt{n}]} a(x,y)\, b(x,y).$$

Choose a prime $q \in [n, 2n]$. Then a and b have unique extensions $\widetilde{a} : \mathbb{F}_q^2 \to \mathbb{F}_q$ and $\widetilde{b} : \mathbb{F}_q^2 \to \mathbb{F}_q$ respectively as degree-$(\sqrt{n}-1)$ polynomials. Also, define the polynomial $s : \mathbb{F}_q \to \mathbb{F}_q$ by

$$s(x) := \sum_{y=1}^{\sqrt{n}} \widetilde{a}(x,y)\, \widetilde{b}(x,y) \pmod{q}.$$

Notice that $\deg(s) \le 2(\sqrt{n}-1)$. Merlin's message to Alice will consist of a polynomial $s' : \mathbb{F}_q \to \mathbb{F}_q$, which also has degree at most $2(\sqrt{n}-1)$, and which is specified by its coefficients. Merlin claims that $s = s'$. If Merlin is honest, then Alice can easily compute the inner product as $IP = \sum_{x=1}^{\sqrt{n}} s(x)$. So the problem reduces to checking that $s = s'$. This is done as follows: first Bob chooses $r \in \mathbb{F}_q$ uniformly at random and sends it to Alice, along with the value of $\widetilde{b}(r,y)$ for every $y \in [\sqrt{n}]$. Then Alice checks that

$$s'(r) = \sum_{y=1}^{\sqrt{n}} \widetilde{a}(r,y)\, \widetilde{b}(r,y) \pmod{q}.$$

If $s = s'$, then the above test succeeds with certainty. On the other hand, if $s \ne s'$, then

$$\Pr_{r \in \mathbb{F}_q}\left[s(r) = s'(r)\right] \le \frac{\deg(s)}{q} \le \frac{1}{3},$$

and hence the test fails with probability at least $\frac{2}{3}$. □

7. CONCLUSIONS AND OPEN PROBLEMS

Arithmetization is one of the most powerful ideas in the history of complexity theory. It led to the IP = PSPACE Theorem, the PCP Theorem, non-relativizing circuit lower bounds, and many other achievements of the last two decades. Yet we showed that arithmetization is fundamentally unable to resolve many of the barrier problems in the field, such as P versus NP, derandomization of RP, and circuit lower bounds for NEXP.

Can we pinpoint what it is about arithmetization that makes it incapable of solving these problems? In our view, arithmetization simply fails to "open the black box wide enough." In a typical arithmetization proof, one starts with a polynomial-size Boolean formula φ, and uses φ to produce a low-degree polynomial p. But having done so, one then treats p as an arbitrary black-box function, subject only to the constraint that $\deg(p)$ is small. Nowhere does one exploit the small size of φ, except insofar as it lets one evaluate p in the first place. The message of this paper has been that, to make further progress, one will have to probe φ in some "deeper" way. To reach this conclusion, we introduced a new model of *algebraic query complexity*, which has already found independent applications in communication complexity, and which has numerous facets to explore in its own right.

We now propose five directions for future work, and list some of the main open problems in each direction.

(1) Find non-algebrizing techniques. This, of course, is the central challenge we leave.

The best example we have today of a non-algebrizing result is arguably the set of cryptographic protocols—including those of Goldreich-Micali-Wigderson [10] and Yao [31]—that exploit the locality of computation in manifestly non-algebraic ways. Yet in the full version of this paper, we show (perhaps surprisingly) that even the GMW protocol algebrizes, assuming the existence of a one-way function that is computable in P (with no oracle) but is secure even against $\mathsf{BPP}^{\widetilde{A}}$ adversaries. It would be interesting to know whether the GMW protocol algebrizes under a more standard cryptographic assumption.

A few other examples of non-relativizing results predating the "interactive proofs revolution" have been proposed. Small-depth circuit lower bounds, such as $\mathsf{AC}^0 \ne \mathsf{TC}^0$ [22], can be shown to fail relative to suitable oracle gates, and are almost certainly non-algebrizing as well. On the other hand, these results are already "well covered" by the natural proofs barrier. In another direction, Arora, Impagliazzo, and Vazirani [2] argue that even the Cook-Levin Theorem (and by extension, the PCP Theorem) should be considered non-relativizing, while Hartmanis et al. [11] make a similar case for the 1977 result of Hopcroft, Paul, and Valiant [13] that $\mathsf{TIME}(f(n)) \ne \mathsf{SPACE}(f(n))$ for any space-constructible f. However, because of subtleties in defining the oracle access mechanism, there is legitimate debate about whether these examples should "truly" be considered non-relativizing; see Fortnow [9] for a contrary perspective.[9]

If arithmetization—which embeds the Boolean field $\mathbb{F}_2$ into a larger field or the integers—is not enough, then a natural idea is to embed $\mathbb{F}_2$ into a non-commutative algebra. But in the full version of this paper, we show that for every subexponential k, the algebra of $k \times k$ matrices still does not suffice. So the question arises: what other useful algebraic structures can mathematics offer complexity theory?

Another potential way around the algebrization barrier is "recursive arithmetization": first arithmetizing a Boolean

[9] Eric Allender has suggested the delightful term "irrelativizing," for results that neither relativize nor fail to relativize.

formula, then reinterpreting the result as a Boolean function, then arithmetizing *that* function, and so on ad infinitum. In the full version of this paper, we show that k-arithmetization is still not powerful enough to prove $\mathsf{P} \neq \mathsf{NP}$, for any constant k. On the other hand, we have no idea whether double-arithmetization is already powerful enough to prove $\mathsf{P} = \mathsf{RP}$ or $\mathsf{NEXP} \not\subset \mathsf{P/poly}$.

(2) Find ways to exploit the *structure* of polynomials produced by arithmetization. This is also a possible way around the algebrization barrier, but seems important enough to deserve its own heading. The question is: given that a polynomial $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$ was produced by arithmetizing a small Boolean formula, *does $\widetilde{A}$ have any properties besides low degree that a polynomial-time algorithm querying it could exploit*? Or alternatively, do there exist "pseudorandom extensions" $\widetilde{A} : \mathbb{F}^n \to \mathbb{F}$—that is, low-degree extensions that are indistinguishable from "random" low-degree extensions by any $\mathsf{BPP}^{\widetilde{A}}$ machine, but that were actually produced by arithmetizing small Boolean formulas?

(3) Find open problems that can still be solved with algebrizing techniques. In the short term, this is perhaps the most "practical" response to the algebrization barrier. Here is a problem that, for all we know, might still be solvable with tried-and-true arithmetization methods: improve the result of Santhanam [26] that $\mathsf{PromiseMA} \not\subset \mathsf{SIZE}\left(n^k\right)$ to $\mathsf{MA} \not\subset \mathsf{SIZE}\left(n^k\right)$.

(4) Prove algebraic oracle separations. Can we show that the interactive protocol of Lund, Fortnow, Karloff, and Nisan [19] cannot be made constant-round by any algebrizing technique? In other words, can we give an oracle A and extension $\widetilde{A}$ such that $\mathsf{coNP}^A \not\subset \mathsf{AM}^{\widetilde{A}}$? In the communication complexity setting, Klauck [17] mentions coNP versus AM as a difficult open problem; perhaps the algebraic query version is easier. The larger challenge is to give algebraic oracles that separate all the levels of the polynomial hierarchy—or at least separate the polynomial hierarchy from larger classes such as $\mathsf{P}^{\#\mathsf{P}}$ and PSPACE.

(5) Understand algebrization better. In defining algebrization, was it essential to give only one machine access to the extension oracle $\widetilde{A}$, and the other access to A? Or could we show (for example) not only that $\mathsf{coNP}^A \subseteq \mathsf{IP}^{\widetilde{A}}$, but also that $\mathsf{coNP}^{\widetilde{A}} \subseteq \mathsf{IP}^{\widetilde{A}}$? Also, low-degree extensions can be seen as just one example of an error-correcting code. To what extent do our results carry over to arbitrary error-correcting codes?

Acknowledgments

We thank Benny Applebaum, Sanjeev Arora, Boaz Barak, Andy Drucker, Lance Fortnow, Russell Impagliazzo, Hartmut Klauck, Adam Klivans, Ryan O'Donnell, Rahul Santhanam, Sasha Sherstov, Amir Shpilka, Madhu Sudan, Luca Trevisan, and Ryan Williams for helpful discussions.

8. REFERENCES

[1] S. Aaronson. Oracles are subtle but not malicious. In *Proc. IEEE Complexity*, p. 340–354, 2006.
[2] S. Arora, R. Impagliazzo, and U. Vazirani. Relativizing versus nonrelativizing techniques: the role of local checkability. Manuscript, 1992.
[3] L. Babai, L. Fortnow, and C. Lund. Nondeterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.
[4] T. Baker, J. Gill, and R. Solovay. Relativizations of the P=?NP question. *SIAM J. Comput.*, 4:431–442, 1975.
[5] H. Buhrman, L. Fortnow, and T. Thierauf. Nonrelativizing separations. In *Proc. IEEE Complexity*, p. 8–12, 1998.
[6] H. Buhrman, N. Vereshchagin, and R. de Wolf. On computation and communication with small bias. In *Proc. IEEE Complexity*, p. 24–32, 2007.
[7] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Comput. Sci.*, 288:21–43, 2002.
[8] A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.
[9] L. Fortnow. The role of relativization in complexity theory. *Bulletin of the EATCS*, 52:229–244, February 1994.
[10] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *J. ACM*, 38(1):691–729, 1991.
[11] J. Hartmanis, R. Chang, S. Chari, D. Ranjan, and P. Rohatgi. Relativization: a revisionistic perspective. *Bulletin of the EATCS*, 47:144–153, 1992.
[12] J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.
[13] J. E. Hopcroft, W. J. Paul, and L. G. Valiant. On time versus space. *J. ACM*, 24(2):332–337, 1977.
[14] A. Juma, V. Kabanets, C. Rackoff, and A. Shpilka. The black-box query complexity of polynomial summation. ECCC TR07-125, 2007.
[15] B. Kalyanasundaram and G. Schnitger. The probabilistic communication complexity of set intersection. *SIAM J. Discrete Math*, 5(4):545–557, 1992.
[16] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55:40–56, 1982.
[17] H. Klauck. Rectangle size bounds and threshold covers in communication complexity. In *Proc. IEEE Complexity*, p. 118–134, 2003.
[18] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31:1501–1526, 2002.
[19] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39:859–868, 1992.
[20] R. Raz. Exponential separation of quantum and classical communication complexity. In *Proc. ACM STOC*, p. 358–367, 1999.
[21] R. Raz and A. Shpilka. On the power of quantum proofs. In *Proc. IEEE Complexity*, p. 260–274, 2004.
[22] A. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$. *Mathematicheskie Zametki*, 41(4):598–607, 1987.
[23] A. A. Razborov. On the distributional complexity of disjointness. *Theoretical Comput. Sci.*, 106:385–390, 1992.
[24] A. A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya Math.*, 67(1):145–159, 2003.
[25] A. A. Razborov and S. Rudich. Natural proofs. *J. Comput. Sys. Sci.*, 55(1):24–35, 1997.
[26] R. Santhanam. Circuit lower bounds for Merlin-Arthur classes. In *Proc. ACM STOC*, p. 275–283, 2007.
[27] A. Shamir. IP=PSPACE. *J. ACM*, 39(4):869–877, 1992.
[28] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991.
[29] N. V. Vinodchandran. A note on the circuit complexity of PP. ECCC TR04-056, 2004.
[30] C. B. Wilson. Relativized circuit complexity. *J. Comput. Sys. Sci.*, 31(2):169–181, 1985.
[31] A. C-C. Yao. How to generate and exchange secrets (extended abstract). In *Proc. IEEE FOCS*, p. 162–167, 1986.