comput. complex. **24** (2015), 393 – 475
© The Author(s). This article is published
with open access at Springerlink.com 2015
1016-3328/15/020393-83
published online May 7, 2015
DOI 10.1007/s00037-015-0103-x

computational complexity

UNIFYING KNOWN LOWER BOUNDS VIA GEOMETRIC COMPLEXITY THEORY

Joshua A. Grochow

Abstract. We show that most algebraic circuit lower bounds and relations between lower bounds naturally fit into the representation-theoretic framework suggested by geometric complexity theory (GCT), including: the partial derivatives technique (Nisan–Wigderson), the results of Razborov and Smolensky on $\mathsf{AC}^0[p]$, multilinear formula and circuit size lower bounds (Raz *et al.*), the degree bound (Strassen, Baur–Strassen), the connected components technique (Ben-Or, Steele–Yao), depth 3 algebraic circuit lower bounds over finite fields (Grigoriev–Karpinski), lower bounds on permanent versus determinant (Mignon–Ressayre, Landsberg–Manivel–Ressayre), lower bounds on matrix multiplication (Bürgisser–Ikenmeyer, Landsberg–Ottaviani) (these last two were already known to fit into GCT), the chasms at depth 3 and 4 (Gupta–Kayal–Kamath–Saptharishi, Agrawal–Vinay, Koiran, Tavenas), matrix rigidity (Valiant) and others. That is, the original proofs, with what is often just a little extra work, already provide representation-theoretic obstructions in the sense of GCT for their respective lower bounds. This enables us to expose a new viewpoint on GCT, whereby it is a natural unification of known results and broad generalization of known techniques. It also shows that the framework of GCT is at least as powerful as previous methods, and gives many new proofs-of-concept that GCT can indeed provide significant asymptotic lower bounds. This new viewpoint also opens up the possibility of fruitful two-way interactions between previous results and the new methods of GCT; we provide several concrete suggestions of such interactions. For example, the representation-theoretic viewpoint of GCT naturally provides new properties to consider in the search for new lower bounds.

Keywords. Lower bounds, circuit complexity, algebraic complexity, geometric complexity theory, representation theory, algebraic geometry.

Subject classification. 68Q17, 14L30, 13A50.

Parts of this paper are written in a survey style in order to make it accessible to a wider audience. In particular, items marked Example or Fact are included only for expository purposes, and we make no claim to originality for those.

This paper presupposes no knowledge of representation theory on the part of the reader. In fact, we use previous lower bounds together with our new viewpoint to motivate the use and definitions of representation theory and algebraic geometry in complexity theory.

1. Introduction

Geometric complexity theory (GCT) is a program toward lower bounds—such as $\mathsf{P} \neq \mathsf{NP}$—using algebraic geometry and representation theory (see Mulmuley (2011b) for an overview, and references therein). In this paper, we show that most algebraic circuit lower bounds naturally fit into the representation-theoretic framework used in GCT. We also show that part of the representation-theoretic approach is necessary, that this approach illuminates lower bounds even when it is not strictly necessary, and that it may in fact be the easiest approach to proving circuit lower bounds. GCT thus provides a unifying framework for many known lower bounds, vastly generalizing known lower bound techniques. This representation-theoretic viewpoint opens the door for new potentially fruitful two-way interactions between previous results and new progress in (geometric) complexity theory (see Sections 1.2 and 4.2 for details).

Essentially, any lower bound proof $\mathcal{C}_{hard} \not\subseteq \mathcal{C}_{easy}$ between nonuniform complexity classes proceeds by finding some "useful" property, which applies to every function in $\mathcal{C}_{easy}$, but not to every function in $\mathcal{C}_{hard}$. The first part of the GCT Program suggests the use of properties of a certain type, namely (linear-)invariant properties defined by the vanishing of polynomials, which we capture in the notion of "separating module" (Definition 2.10). Recall that a property Π is linear-invariant if for every function on n variables, $f(\mathbf{x})$ has Π if and only if $f(A\mathbf{x})$ has Π for every invertible $n \times n$ change of variables A. As an example of a property that is defined by polynomials: The property of $ax^2 + bxy + cy^2$ being a perfect square is equivalent to (or defined by) the vanishing of the polynomial $b^2 - 4ac$ (see Section 2.1 for a more leisurely explanation).

In this paper, we show that most known algebraic circuit lower bounds in fact use separating modules, including:

- Lower bounds on restricted depth 3 algebraic circuits in characteristic zero (Nisan & Wigderson 1996/97),
- Lower bounds on unrestricted depth 3 algebraic circuits over finite fields (Grigoriev & Karpinski 1998),
- The recent lower bounds on depth 4 algebraic circuits with bottom fan-in $O(\sqrt{n})$ (Gupta, Kamath, Kayal & Saptharishi 2012),
- Lower bounds on multilinear formula size (Raz 2009),
- The degree bound of Strassen (1972/73) and Baur & Strassen (1983) (see below),
- Lower bounds on real (semi-)algebraic decision trees (Ben-Or 1983; Yao 1997),
- "$\mathsf{P} \neq \mathsf{NC}$" in the PRAM model without bit operations (Mulmuley 1999),
- Lower bounds on bounded depth Boolean circuits (Razborov 1987; Smolensky 1987),
- The best known lower bounds on permanent versus determinant (Mignon & Ressayre (2004), already shown to use a separating module in Landsberg, Manivel & Ressayre (2013)),

- Many lower bounds on matrix multiplication (already shown to use a separating module, as in Bürgisser & Ikenmeyer (2013); Landsberg & Ottaviani (2011); Strassen (1987)).

We expect that results which use similar techniques can be shown to use separating modules as well, such as Raz (2006), Raz *et al.* (2008), Raz & Yehudayoff (2009), Shpilka & Wigderson (2001), Grigoriev & Razborov (2000), Yao (1991), Björner *et al.* (1992). We also observe that many relations between lower bounds yield relations between separating modules. In other words, a separating module that implies lower bound A yields a separating module that implies lower bound B:

- Lower bounds on partial derivatives imply circuit lower bounds (Baur & Strassen 1983),

- Matrix rigidity implies circuit lower bounds (Valiant 1977),

- The chasm at depth 4 (Agrawal & Vinay 2008; Koiran 2012; Tavenas 2013) and the recent chasm at depth 3 (Gupta, Kamath, Kayal & Saptharishi 2013; Tavenas 2013),

- Tensor-rank lower bounds imply formula size lower bounds (Raz 2010b).

Finally, in Section 3 we argue that the use of invariant properties is essentially necessary and that the use of separating modules is the easiest way to prove algebraic circuit lower bounds. Thus, separating modules are the first approach to try and, indeed, may be the only approach that is easy enough that it will ever be carried out. We can already give one such argument: Most algebraic circuit lower bounds already use separating modules.

This new viewpoint makes new tools available and suggests new conjectures and directions to better understand complexity classes and lower bounds. We do not provide new proofs of any of the above results, but rather we offer a meta-observation about many lower bounds, analogous to Natural Proofs (Razborov & Rudich 1997) or the papers Razborov (1995a,b) on bounded arithmetic. This involves digging into the details of the proofs of known lower bounds to understand them in a particular way, which is sometimes trivial but sometimes requires new insights. These previous meta-results have shown that a new viewpoint can be very fruitful: For example, by working in the framework of bounded arithmetic, Razborov was able to come up with a beautiful new proof of the Switching Lemma (Razborov 1995a). Despite this new proof of a lower bound against AC^0, the fundamental message of the papers Razborov & Rudich (1997), Razborov (1995a,b) was negative, giving barriers to proving strong lower bounds, whereas the message of this paper is *positive*, suggesting a *route to proving lower bounds*—a route that most algebraic circuit lower bounds have already begun to traverse.

1.1. Outline. In Section 1.2, we discuss some of the implications of this work. We postpone further details of the implications until Section 4, as they are difficult to discuss properly without definitions and a full example in mind. We give the definitions and an example of how a previous lower bound fits into this new viewpoint in Section 2. In Section 3 and Appendix B, we argue for the necessity of invariant properties and the feasibility and utility of separating modules, especially in comparison with other possible approaches. Section 4 contains further discussion and implications. We discuss the relation of this viewpoint to the larger GCT Program; in particular, separating modules are only the very beginning of the GCT approach. We also discuss which lower bounds do not seem to fit into this framework—mostly those based on uniform hierarchy theorems—and we suggest some concrete directions for future research to push forward both our understanding of GCT and our understanding of known lower bounds and the complexity classes they consider. We also discuss in what way Boolean lower bounds fit into this framework. In Sections 5 and 6, we prove that the results mentioned above use separating modules. However, if the reader is willing to take the above lists on faith, the significance of this paper can be understood without reading these last two sections in detail.

1.2. Implications. Our unifying viewpoint suggests the possibility of a fruitful two-way interplay between the methods currently being leveraged in GCT against major open problems like permanent versus determinant and P versus NP, and already hard-won knowledge for lower bounds on more tractable problems. Although we can state some of these possible interactions now, they will become clearer after the example in the next section, and we discuss further implications in Section 4.

First, the representation-theoretic viewpoint suggests where to look for new properties that might yield lower bounds. Even for lower bounds that are already essentially tight, the representation theory suggests how we might get new proofs of these lower bounds or otherwise understand them better.

Second, the representation-theoretic viewpoint suggests new conjectures, directions, and techniques that may prove fruitful; see, for example, the last paragraph of Section 4.1 and the open questions in Sections 4.2 and 4.6.

Third, by showing that previous lower bounds and GCT share a common representation-theoretic viewpoint, we reveal many new contexts in which it might hopefully be easier to develop the tools and techniques of algebraic geometry and representation theory needed for the GCT approach to bigger problems such as permanent versus determinant or P versus NP.

Fourth, it is often asked how difficult it is to reprove known lower bounds using GCT. The viewpoint in this paper reveals that most of the old proofs *already* give representation-theoretic knowledge crucial to the GCT approach, in the form of separating modules. There is, however, a difference between separating modules and the geometric obstructions defined in Mulmuley & Sohoni (2008). Upgrading the previous lower bounds to yield such geometric obstructions is one of the open questions we discuss in more detail in Section 4.1. This is one of the ways in which GCT suggests how we might understand previous lower bounds better, even ones that are essentially tight.

For now, we mention just one more point: The representation-theoretic viewpoint replaces the amorphous notion of "useful property" with the specific mathematical notion of separating module. In Section 3, we show that this is without loss of generality, so long as one is proving lower bounds by the polynomial method (which we show that essentially all known bounds do). This reduces an amorphous search for new useful properties to a comparatively feasible search for separating modules, which can even be made computational (see Appendix B.3 and Section 4 for more).

2. Definitions and a motivating example

Most non-uniform lower bounds $\mathcal{C}_{hard} \not\subseteq \mathcal{C}_{easy}$ are proved by finding a property shared by all functions in the "easy" class $\mathcal{C}_{easy}$ that some function $f \in \mathcal{C}_{hard}$ does not have. The goal of this section is to introduce a representation-theoretic formalization of the types of properties used by most algebraic circuit lower bounds, namely (linear-)invariant properties defined by polynomials.

2.1. Properties defined by polynomials. Throughout this section, we use the running example of the space $\mathrm{Poly}^2(x, y) = \{ax^2+bxy+cy^2 | a, b, c \in \mathbb{F}\}$ of degree 2 homogeneous polynomials in two variables over some field $\mathbb{F}$,[1] and the expression $b^2 - 4ac$.[2] The space $\mathrm{Poly}^2(x, y)$ in this running example should be thought of as a toy version of the space of polynomials we care about, like the determinant, permanent, which are points in $\mathrm{Poly}^n(x_{11}, x_{12}, \ldots, x_{nn})$, but is small enough that we can carry out computations completely by hand, and the definitions in this context should already be familiar to the reader.

Recall that $b^2 - 4ac = 0$ if and only if $ax^2 + bx + c = 0$ has a double root; equivalently, by homogenizing with y, $b^2 - 4ac = 0$ if and only if $ax^2 + bxy + cy^2$ is a perfect square $(\delta x + \eta y)^2$ for some constants $\delta, \eta \in \mathbb{F}$. We thus view $b^2 - 4ac \overset{?}{=} 0$ as a test for the property of being a perfect square, and we say that this *property is defined by the (vanishing of the) polynomial* $b^2 - 4ac$.

Note that here we consider $b^2 - 4ac$ not just as an expression, but as a polynomial in the *variables* a, b, c, which are the coeffi-

[1] In some of these examples, it may be necessary to restrict the characteristic of the field. In all of our actual results, we specify the field more carefully.

[2] The notation $\mathrm{Poly}^d(x_1, \ldots, x_n)$ is not standard. We use it because it is clear and mnemonic. For reference, we give the standard notation from the literature in Appendix D.

cients of the polynomials $ax^2 + bxy + cy^2$. In symbols, $b^2 - 4ac \in \mathrm{Poly}^2(a, b, c) = \mathrm{Poly}^2(\mathrm{Poly}^2(x, y))$. Because there are two different spaces of polynomials here, we find it useful to give different names to them. We refer to polynomials such as $\alpha x^2 + \beta xy + \gamma y^2 \in \mathrm{Poly}^2(x, y)$ with α, β, γ constants as *input polynomials*: These are polynomials in the "input variables" x, y and are also themselves inputs for the property tests. We refer to polynomials such as $b^2 - 4ac \in \mathrm{Poly}^2(\mathrm{Poly}^2(x, y))$ as *test polynomials*: These are polynomials whose variables are the *coefficients* of the input polynomials and define a test for some property of input polynomials.

We index monomials by their exponent vectors $e \in \mathbb{Z}^n_{\geq 0}$ and write $\mathbf{x}^e \overset{def}{=} x_1^{e_1} \dots x_n^{e_n}$; we denote the corresponding coefficient by α_e and the corresponding test variable by a_e, and then write any polynomial as $f(\mathbf{x}) = \sum_{e \in \mathbb{Z}^n_{\geq 0}} \alpha_e \mathbf{x}^e$ (only finitely many terms will be nonzero). That is, $a_e(f) = \alpha_e$. If $p \in \mathbb{F}[(a_e)_{e \in \mathbb{Z}^n_{\geq 0}}]$ is a test polynomial and $f = \sum_e \alpha_e \mathbf{x}^e$ is an input polynomial, we write $p(f)$ for the evaluation of p in which each test variable a_e is set to the corresponding coefficient $\alpha_e \in \mathbb{F}$ of f.

Definition 2.1. *A property* Π *of input polynomials is* defined by (test) polynomials *if there is a set of test polynomials* $p_1, \dots, p_k$ *such that* $f(\mathbf{x})$ *has property* Π *if and only if* $p_1(f) = p_2(f) = \dots = p_k(f) = 0$.

Remark 2.2. *Readers familiar with algebraic geometry will note that a property defined by test polynomials is exactly the same thing as an algebraic subset of the vector space* $\mathrm{Poly}^d(x_1, \dots, x_n)$ *of input polynomials. This is an algebro-geometric viewpoint on complexity. We discuss this further in* Section 3. *For now we note that such algebro-geometric notions of complexity have been used before: Border rank for matrix multiplication and "infinitesimal approximation" in GCT are both algebro-geometric notions of complexity in this sense.*

Remark 2.3. *By Hilbert's Basis Theorem, any property defined by polynomials can be defined by finitely many polynomials.*

2.2. Linear-invariant properties defined by polynomials. Kayal (2011, Sec. 5.2) observed that several lower bounds use linear-invariant properties at their core, and in fact, this observation was the starting point for this paper. In this paper, we extend this observation in two directions simultaneously: (1) We observe that most algebraic circuit lower bounds use (linear-)invariant properties *defined by polynomials* (Definition 2.1), allowing us to make the connection with representation theory and GCT, and (2) we extend the observation to most algebraic, and some Boolean, circuit lower bounds.

Definition 2.4. *A property* Π *of (input) polynomials is* linear-invariant *if for every polynomial* $f(x_1, \ldots, x_n)$ *and every invertible linear change of variables* $A \in \mathrm{GL}_n(\mathbb{F})$

$$f(\mathbf{x}) \textit{ has property } \Pi \iff f(A\mathbf{x}) \textit{ has property } \Pi.$$

Example 2.5. The property of being a perfect square is linear-invariant: $f(\mathbf{x}) = g(\mathbf{x})^2$ if and only if $f(A\mathbf{x}) = g(A\mathbf{x})^2$ for any invertible linear change of variables A. As explained in the previous section, in the case of $f(x, y)$ homogeneous of degree 2, this property is defined by the vanishing of the test polynomial $b^2 - 4ac$. ◊

Example 2.6. The dimension of the space of all partial derivatives of a homogeneous polynomial f is a linear-invariant property. The k-th order partial derivatives of f are linearly independent from its ℓ-th order partial derivatives for $k \neq \ell$, so we may prove this for each k separately. Consider the partial derivative $\left(\frac{\partial f}{\partial x_{i_1} \cdots \partial x_{i_k}}\right)(\mathbf{x})$. When we transform the variables $\mathbf{x}$ by A, we change both the variables with respect to which the derivatives are being taken, and we change the variables at which the partial derivative is being evaluated. The fact that the former kind of transformation does not change the dimension of the space of partial derivatives follows from the usual "directional derivative" formula from multivariate calculus. The latter kind of transformation also does not change the dimension of a space of polynomials, for $\sum_{i=1}^{d} \alpha_i g_i(\mathbf{x}) = 0$ if and only if $\sum_{i=1}^{d} \alpha_i g_i(A\mathbf{x}) = 0$. We will see below that this property is also defined by polynomials. $\Diamond$

The notion of a linear-invariant property defined by polynomials is embodied in the following definition. To make the definition clear, we first introduce one more bit of notation. Each linear change of input variables $B \in \mathrm{GL}_n(\mathbb{F})$ defines a linear map Coeff_B from $\mathrm{Poly}^d(x_1, \dots, x_n)$ to itself: B sends $f(\mathbf{x}) = \sum_e \alpha_e \mathbf{x}^e$ to $(B \cdot f)(x) \overset{def}{=} f(B\mathbf{x}) = \sum_e \alpha'_e \mathbf{x}^e$, and Coeff_B is thus the linear map taking the coefficient vector $(\alpha_e)_{e \in \mathbb{Z}^n_{\geq 0}}$ to the new coefficient vector $\mathrm{Coeff}_B((\alpha_e)_e) = (\alpha'_e)_e$. It is a standard fact—easily verified—that Coeff_B is linear in the coefficients α_e. Thus, B induces a linear map Coeff_B on the coefficients of input polynomials, which are in turn the variables of test polynomials. Then Coeff_B induces a linear map on test polynomials, taking $p((a_e)_e)$ to $p(\mathrm{Coeff}_B((a_e)_e))$.

Definition 2.7. *A* test $\mathrm{GL}_n(\mathbb{F})$-module *is a finite-dimensional vector space T of test polynomials, such that for each $p \in T$ and each $B \in \mathrm{GL}_n(\mathbb{F})$, $p(\mathrm{Coeff}_B((a_e)_e))$ also lies in T.*[3]

Note that if $\{p_1, \dots, p_k\}$ is a set of test polynomials, then to check that the linear span T of the p_i is a test module, it suffices to check that $p_i \circ \mathrm{Coeff}_B$ is in T for each basis polynomial and each $B \in \mathrm{GL}_n(\mathbb{F})$.[4] In practice, such checks are rarely necessary, because of the following equivalence.

We say a test module T vanishes on an input polynomial f if every test polynomial $p \in T$ vanishes at f. Conversely, we say T does not vanish on f if there exists a test polynomial $p \in T$ that does not vanish at f. The set of input polynomials at which a given test module vanishes is a linear-invariant set, which we can think of as a linear-invariant property:

[3] See Appendix C for a discussion of the terminology.

[4] In fact, one need only consider those B in a generating set for $\mathrm{GL}_n(\mathbb{F})$. When $\mathbb{F}$ is a finite field, there are finite generating sets. When $\mathbb{F}$ is infinite, one can instead consider a finite generating set of the Lie algebra of $\mathrm{GL}_n(\mathbb{F})$. This technique is standard in the representation theory of GL_n, as in Fulton & Harris (1991), but is beyond the scope of this paper.

FACT 2.8. *There is an infinite-to-one correspondence between test* $\mathrm{GL}_n(\mathbb{F})$*-modules and linear-invariant properties defined by polynomials.*

That is, each linear-invariant property defined by polynomials is defined by some test $\mathrm{GL}_n(\mathbb{F})$-module, and each test $\mathrm{GL}_n(\mathbb{F})$-module defines a linear-invariant property.[5] The proof involves only basic observations regarding group actions and algebraic sets (see Appendix A).

EXAMPLE 2.9. The vector space spanned by the test polynomial $b^2 - 4ac$ is a one-dimensional test $\mathrm{GL}_2(\mathbb{F})$-module. For let $f(x, y) = ax^2 + bxy + cy^2$ and $A = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$, and write $f(A\mathbf{x}) = a(\alpha x + \beta y)^2 + b(\alpha x + \beta y)(\gamma x + \delta y) + c(\gamma x + \delta y)^2 = a'x^2 + b'xy + c'y^2$. Let $p(a, b, c) = b^2 - 4ac$; then $p(\mathrm{Coeff}_A(a, b, c)) = p(a', b', c') = b'^2 - 4a'c'$. A simple but tedious calculation then reveals that $b'^2 - 4a'c' = \det(A)^2\,(b^2 - 4ac)$ and hence that $p(\mathrm{Coeff}_A(a, b, c))$ is a scalar multiple of $p(a, b, c)$. ◊

2.3. Separating modules and a first example.

DEFINITION 2.10. *A* separating module *for a lower bound* $\mathcal{C}_{hard} \not\subseteq \mathcal{C}_{easy}$ *is a test module* T *such that* T *vanishes on every function in* $\mathcal{C}_{easy}$*, but does not vanish at some function* $f_{hard} \in \mathcal{C}_{hard}$*.*[6]

The main thesis of this paper is that most algebraic circuit lower bounds already use separating modules. We now demonstrate this with an example, by showing that Theorem 0 of Nisan & Wigderson (1996/97) uses a separating module. We first recall their definitions and result. In the next section, we show that the existence of a separating module was necessary, assuming that the bound was provable by test polynomials at all (Lemma 3.1).

[5] If $\mathbb{F}$ is algebraically closed, then Hilbert's Nullstellensatz implies that two test modules T_1, T_2 define the same invariant property if and only if the ideals of test polynomials generated by T_1 and T_2 have the same radical. Recall that the radical of an ideal I is the ideal $\sqrt{I} \overset{def}{=} \{f : f^k \in I \text{ for some } k \geq 1\}$.

[6] Separating modules are nearly equivalent to the "HWV obstructions" of Bürgisser & Ikenmeyer (2013). For a discussion of the exact relationship and choice of terminology, see Appendix C.

An algebraic circuit is *homogeneous* if every gate in the circuit computes a homogeneous polynomial. The d-th elementary symmetric function in n variables, denoted $e_{d,n}$, is the sum of all multilinear monomials of degree d.

THEOREM 2.11 (Nisan & Wigderson 1996/97, Theorem 0). *Over a field of characteristic zero, any homogeneous depth 3 algebraic circuit computing $e_{2d,n}$ has size $\Omega\left(\left(\frac{n}{4d}\right)^d\right)$.*

When $d = cn$ for any $0 < c \leq 1/4$, this lower bound is exponential in n.

PROOF OUTLINE. The key property they consider is the dimension of the space of all partial derivatives (of all orders) of a function. We denote this space $\partial(f)$. First, they show that $\dim \partial(C) \leq s2^d$ for any homogeneous depth 3 algebraic circuit C of size s computing a polynomial of degree d. Next, they show that the dimension of $\partial(e_{2d,n})$ is at least $\binom{n}{d}$; this is the only part of the proof that seems to depend on the field having characteristic zero. Combining these inequalities, one gets $s2^{2d} \geq \binom{n}{d} \geq \left(\frac{n}{d}\right)^d$. □

The following proposition was independently shown, in slightly different language, in Landsberg (2014a, Section 8.1).

PROPOSITION 2.12. *There is a separating module for the lower bound of Theorem 2.11.*

PROOF. Let $\Pi(r)$ denote the property "$\dim \partial(f) \leq r$." We argued in Example 2.6 that $\dim \partial(f)$ is a linear-invariant property for homogeneous f. We now show that this property is defined by a test $\mathrm{GL}_n(\mathbb{F})$-module and hence that the above proof yields a separating module.

Let $f(\mathbf{x}) = \sum_e \alpha_e \mathbf{x}^e$ be a homogeneous polynomial of degree d (the only nonzero terms in the sum are those for which $\sum_i e_i = d$) and consider the following matrix M_f. The columns of M_f are indexed by the monomials of degree $\leq d$, and the rows of M_f are indexed by the partial derivative operators (these are in bijective correspondence with monomials, but we refer to them this way to keep track of which is which). The entry in the $\partial^k/\partial x_{i_1} \cdots \partial x_{i_k}$ row and the $\mathbf{x}^e$ column is the coefficient of $\mathbf{x}^e$ in $\partial^k f/\partial x_{i_1} \cdots \partial x_{i_k}$. Note that this coefficient is some linear combination of the coefficients a_e of f.

Then the dimension of $\partial(f)$ is the same as the (row) rank of M_f. It is a standard fact from linear algebra that M_f has rank $\leq r$ if and only if all the $(r+1) \times (r+1)$ minors of M_f vanish. Each such minor is a degree $r+1$ polynomial of the entries of M_f, which are themselves linear combinations of the coefficients a_e of f. Hence, each such minor is a test polynomial of degree $r+1$. Let $T(r)$ denote the linear span of these minors. We have just shown that (the vanishing of the test polynomials in) $T(r)$ defines the property $\Pi(r)$.

In particular, $\Pi(r)$ is a linear-invariant property defined by polynomials. By Fact 2.8, $\Pi(r)$ is defined by some test module, which is thus a separating module. However, we can argue further that $T(r)$ itself is a test $\mathrm{GL}_n(\mathbb{F})$-module, and hence a separating module for the lower bound of Theorem 2.11.

In Example 2.6, we essentially showed that $M_{f(A\mathbf{x})}$ is related to $M_{f(\mathbf{x})}$ by left and right multiplication by some matrices related to A (in a similar way to how Coeff_A is related to A). It is a standard fact about minors that the $(r+1) \times (r+1)$ minors of BM_fC are linear combinations of the $(r+1) \times (r+1)$ minors of M_f. Hence for any test polynomial $p \in T(r)$, $p \circ \mathrm{Coeff}_A$ is also in $T(r)$. Thus, $T(r)$ is a separating module for Theorem 2.11. □

As with everything in complexity, in fact what we have is a family of separating modules. Namely, if we consider $e_{2d,n}$ with $d = n/8$, then $T(2^{3n/8})$ vanishes at every polynomial computed by a depth 3 homogeneous circuit of degree $n/4$ and size at most $2^{n/8}$, but does not vanish at $e_{n/4,n}$.

2.4. Generalizations. For other lower bounds, it is useful to generalize some of the above notions.

First, we allow input objects other than polynomials. For example, in the context of matrix rigidity it will be useful to consider input matrices. Regardless of the input objects, we still speak of test polynomials. In the case of input matrices, test polynomials are then polynomials whose variables are the coordinates a_{ij} of the input matrices. In the context of Boolean functions, we often first represent a function by its unique multilinear polynomial and then work in the context of input polynomials. In the context of the degree bound (Baur & Strassen 1983; Strassen 1972/73) and the connected components sorting lower bound (Ben-Or 1983), the input objects are (semi-)algebraic sets, given by their defining polynomial (in)equalities. The variables for the test polynomials are then the coefficients of the equations defining the (semi-)algebraic sets.

Second, we can allow other types of invariance besides linear invariance. Another type of invariance which we will use in the context of matrix rigidity and multilinear lower bounds is *permutation-invariance*: $f(x_1, \ldots, x_n)$ has the property if and only if $f(x_{\pi(1)}, \ldots, x_{\pi(n)})$ has the property, for any permutation π. We then speak of test S_n-modules, and the analog of Fact 2.8 holds (Fact A.1).

Affine invariance also arises frequently. Here, we generalize from linear transformation $\mathbf{x} \mapsto A\mathbf{x}$ to affine transformations $\mathbf{x} \mapsto A\mathbf{x}+\mathbf{b}$, with $A \in \mathrm{GL}_n(\mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^n$. The group of all such transformations is the affine general linear group $\mathrm{AGL}_n(\mathbb{F})$. We then speak of *affine-invariant* properties and test $\mathrm{AGL}_n(\mathbb{F})$-modules. Again, the analog of Fact 2.8 holds (Fact A.1). When the invariance is understood from context, we may simply refer to test modules and separating modules without reference to a particular group.

3. On the necessity and utility of separating modules and border complexity

Here, we show that the use of invariant properties for lower bounds is necessary. By Fact 2.8, to show that separating modules are necessary it would then suffice to show that the use of test polynomials is necessary. Although this is not strictly true for all lower bounds, in Appendix B we discuss situations where this is true (e.g., matrix multiplication), and we argue that even when the use of test polynomials is not strictly necessary, it still provides useful information and is likely the easiest approach to try. Although we feel that this second half of the argument is important for putting various approaches to lower bounds in their appropriate context, we defer it to an appendix only because it is heuristic, somewhat technical, and possibly contentious, and we do not wish to distract from the main points of the paper. However, one argument for this which we can already state is that *most algebraic circuit lower bounds already use separating modules*, as shown in this paper.

First we show that if $\mathcal{C}_n$ is invariant under some group G—such as GL_n, S_n—then any property used to prove a lower bound against $\mathcal{C}_n$ can be transformed into a G-invariant property that proves the same lower bound. Then we show that essentially all "naturally occurring" complexity classes and complexity measures are permutation-invariant, and many are linear- or affine-invariant. Finally, we discuss in what sense complexity classes are naturally objects of algebraic geometry, foreshadowing the more in-depth discussion in Appendix B.

The use of algebraic geometry in computational complexity should thus be seen as *completely natural*, in the same sense that it would be completely natural to use linear algebra to understand complexity classes if it turned out they were all vector spaces, ring theory if they were all rings, or topology if they were all topological spaces.

Terminological note for readers familiar with invariant theory: when we write of a "G-invariant set" such as a complexity class or property, we mean that it is setwise G-invariant: the action of G may permute the set, but only among its own elements. This is distinct from sets defined by the vanishing of a G-invariant test polynomial; the latter indeed define G-invariant sets, but there are many G-invariant sets that cannot be defined by a single G-invariant test polynomial. A G-invariant test polynomial is a one-dimensional test G-module, and is very unlikely to be a separating module in any case of interest. Part of the point of using such potentially confusing terminology (for those familiar with invariant theory) is that it makes it clear that test G-modules are simply the natural generalization from sets defined by G-invariant test polynomials (the fundamental objects of invariant theory) to general G-invariant sets.

Throughout this section and Appendix B, we only discuss non-uniform lower bounds. If $\mathcal{C}$ is a non-uniform complexity class, then $\mathcal{C}_n$ denotes the functions in $\mathcal{C}$ with n inputs. By a "property" in general, we mean a set of input polynomials, or more generally input objects.

LEMMA 3.1. *If any property can be used to proved a lower bound against a G-invariant complexity class, then a G-invariant property can be used to prove the same lower bound.*

PROOF. Suppose property Π is used to prove a lower bound against $\mathcal{C}_n$ by showing that $\mathcal{C}_n \subseteq \Pi$ and $f_{hard,n} \notin \Pi$.[7] Let Π^G denote the unique inclusion-maximal G-invariant subset contained in Π; this exists by Zorn's Lemma, as an arbitrary union of G-invariant subsets is G-invariant. As $\mathcal{C}_n$ is G-invariant, by the definition of Π^G we have $\mathcal{C}_n \subseteq \Pi^G$. The G-invariant property Π^G then proves the same lower bound as Π, as $f_{hard,n} \notin \Pi \supseteq \Pi^G \supseteq \mathcal{C}_n$. □

OBSERVATION 3.2. *Nearly all complexity classes are permutation-invariant.*

All complexity measures and (non-uniform) complexity classes we are aware of are permutation-invariant: They do not depend on the names or order of the variables.[8] Indeed, we imagine that any complexity class or measure that was *not* permutation-invariant would be perverse, as the (non-uniform) complexity of computing a function should really not depend on whether its variables are called $x_1, \dots, x_n$ or $a, b, c, \dots,$ or $x_n, \dots, x_1$. Thus we can expect that any lower bound uses a permutation-invariant property, at the very least.

[7] For readers familiar with Natural Proofs (Razborov & Rudich 1997), note that we are using the complementary notion of "useful property" here. They use properties Π that are disjoint from $\mathcal{C}_n$, whereas we use properties Π that completely contain $\mathcal{C}_n$. By taking the complements of sets, the two viewpoints are equivalent. We chose our viewpoint because it has nicer algebro-geometric properties, as in Appendix B.3.

[8] The only exceptions we are aware of are in uniform complexity, namely regular and context-free languages. However, the size of the non-uniform analog of regular languages—read-once branching programs—is permutation-invariant. We thank Michael Forbes for these examples.

Many complexity classes, particularly algebraic ones, are furthermore linear- or affine-invariant. For example, algebraic circuit size does not change by more than an additive difference of n (or n^2, depending on the exact model) after a linear or affine transformation. Additionally, circuit depth increases by at most 1; for circuits whose bottom gates are linear combination gates, the depth need not increase at all. For example, $\mathsf{AC}^0[2]$ is $\mathrm{AGL}_n(\mathbb{F}_2)$-invariant (though we note that AC^0 is not $\mathrm{GL}_n(\mathbb{F}_2)$-invariant, as $\mathbb{F}_2$-linear transformations are as powerful as parity). This is in line with Kayal's initial observation (Kayal 2011, Sec. 5.2) that several known lower bounds use affine-invariant properties, and with our observations in this paper.

Combining Lemma 3.1 and Observation 3.2: for all naturally occurring non-uniform complexity classes, if any property can be used to prove a lower bound, a permutation-invariant property can be used.

Non-uniform complexity classes are algebro-geometric objects. A complexity class $\mathcal{C}_n$ is typically not defined by the vanishing of some test polynomials. Hence when we prove a lower bound against $\mathcal{C}_n$ using test polynomials, we in fact prove a lower bound against the slightly larger class which we denote $\overline{\mathcal{C}_n}$ and refer to as "border-$\mathcal{C}_n$," in line with normal usage in other contexts (the overline is for Zariski closure; see Definition B.4). Standard results in algebraic geometry (e.g., Mumford 1976, Theorem 2.33; Bürgisser, Clausen & Shokrollahi 1997, Section 20.6) imply that $\overline{\mathcal{C}_n}$ consists of all functions f which can be written as a limit of functions in $\mathcal{C}_n$.[9]

In Appendix B.2, we prove that essentially all non-uniform complexity classes are nonetheless "constructible" by test polynomials, and hence are objects squarely in the domain of algebraic geometry. It is common in algebraic geometry to study such constructible sets by instead studying their Zariski closures, that is, using test polynomials. In Appendix B.3, we give further arguments to support

[9] Over $\mathbb{C}$ the notion of limit is defined in the usual manner. Over, say, $\overline{\mathbb{F}}_p$, we say a function $f(x)$ is a limit of points in $\mathcal{C}_n$ if there is a function $F(t,x)$ that is a formal power series in t with coefficients in $\overline{\mathbb{F}}_p[x]$ such that $F(0,x) = f(x)$ and such that $F(t,x)$ satisfies any test polynomial that vanishes everywhere on $\mathcal{C}_n$.

the utility of test polynomials—and hence separating modules, by Lemma 3.1—in the context of lower bounds.

4. Discussion, relation to the GCT Program, and future directions

In this paper, we show that most algebraic lower bounds and implications between such bounds naturally fit into the representation-theoretic framework suggested by geometric complexity theory, specifically in the form of separating modules. In this section, we discuss further implications of this connection, as well as which lower bounds seem not to fit into this framework (ones which are essentially uniform), the status of lower bounds in positive characteristic, and the relation between this work and the larger GCT Program.

In this short space, and with the level of background assumed in this paper, we will not be able to cover all the relevant aspects of the GCT Program. However, we do endeavor to show how separating modules can be used as an entry point for understanding much of the work in GCT. In particular, it turns out that the *boundaries* of complexity classes—$\overline{\mathcal{C}} \backslash \mathcal{C}$—play an important role that we will just barely be able to touch on in Appendix B. More on this particular topic can be found in several papers (e.g., Kumar 2013; Landsberg 2014a; Landsberg, Manivel & Ressayre 2013; Mulmuley & Sohoni 2001, 2008).

Eric Allender observed that all the lower bounds mentioned here use natural properties in the sense of Razborov & Rudich (1997),[10] and asked whether this was just a coincidence. In light of the generality of separating modules (Section 3), we believe that it is indeed a coincidence and has more to do with the fact that most known results use such properties than it has to do with any inherent limitations of the representation-theoretic viewpoint.

[10] The Boolean properties satisfy the Razborov–Rudich conditions. Although there is no known algebraic analog of the Razborov–Rudich barrier, the algebraic properties mentioned in the previous sections seem like they ought to fulfill the requirements of such an analog, were it to exist. It is worth mentioning that Aaronson & Drucker (2009) have some initial results in the direction of a theory of algebraic natural proofs.

Indeed, there is evidence that the GCT Program over $\mathbb{C}$ avoids the Razborov–Rudich barrier[10] (Grochow 2012, Sec. 3.4.3 gives an overview of such evidence).

4.1. Relation to the geometric complexity theory program.

To state how the separating modules used in this paper differ from the geometric obstructions defined by Mulmuley & Sohoni (2008), and to discuss possible further interactions between previous results and geometric complexity theory, we recall two standard definitions from representation theory, as applied to test modules.

DEFINITION 4.1. *Two (test) G-modules T_1, T_2 are* (linearly) equivalent *if there is a bijective linear map $L\colon T_1 \to T_2$ such that for all $A \in \mathrm{GL}_n$ and all test polynomials $p \in T_1$, $L(p \circ \mathrm{Coeff}_A) = L(p) \circ \mathrm{Coeff}_A$.*[11]

A (test) G-module T (or its linear equivalence class) is irreducible *if there is no nonzero proper subspace of T that is also a (test) G-module.*

A classical theorem (see, e.g., Fulton & Harris 1991) says that over an algebraically closed field of characteristic zero, every GL_n- or S_n-module is a direct sum (as vector spaces) of irreducible submodules. In particular, this implies that if there is a separating module for a lower bound over $\mathbb{C}$, there is an irreducible separating module. We could have included irreducibility in the definition of test module for this reason, but chose not to in order to keep the definition simple and to avoid complications over other fields,

[11] The definition of linear equivalence only remembers the action of G on the test modules, but forgets the fact that the test modules consist of polynomials, or equivalently forgets the corresponding properties defined by those modules. In this sense, the definition is "purely representation-theoretic." Indeed, generically speaking two linearly equivalent test modules can define distinct G-invariant properties, and conversely even two test G-modules that are not linearly equivalent can define the same G-invariant property (see Footnote 5 on page 403). Nonetheless, as the definition of "equivalent" is completely standard in representation theory, we use it here. We add the adjective "*linearly* equivalent" (not typically used in representation theory) to emphasize that we are only considering the G-linear structures of the test modules and not the polynomials they are made up of nor properties they define.

especially finite fields. The property of splitting into a direct sum of irreducible submodules ("completely reducibility") is known to fail in general for AGL_n-modules, even over $\mathbb{C}$, and for GL_n- and S_n-modules in positive characteristic.

To discuss the geometric obstructions of GCT, we work over $\mathbb{C}$. By complete reducibility, the space of all test polynomials can be written as a direct sum of irreducible test $\mathrm{GL}_n(\mathbb{C})$-modules. If we group these modules by their equivalence classes, we may write the space of all test polynomials as the direct sum $\bigoplus_\lambda \bigoplus_{i=1}^{m_\lambda} T_{\lambda,i}$ where the λs index the irreducible equivalence classes. It turns out that each equivalence class λ can only occur among a specific degree $d(\lambda)$ of test polynomials, and since the space of test polynomials of any fixed degree is finite-dimensional, each m_λ is finite. Furthermore, the numbers m_λ are independent of the choice of direct sum. We refer to m_λ as the *multiplicity* of the equivalence class λ in the space of test polynomials.

If $\mathcal{C}$ is a linear-invariant complexity class, then we may divide up the test modules into those which vanish everywhere on $\mathcal{C}$ (and hence on $\overline{\mathcal{C}}$, see Appendix B.1), and those which do not vanish somewhere on $\mathcal{C}$. We say that a test module which does not vanish somewhere on $\mathcal{C}$ is *supported* on $\mathcal{C}$. Let $m_\lambda(\overline{\mathcal{C}})$ denote the number of test modules in the equivalence class λ that are supported on $\overline{\mathcal{C}}$.

Definition 4.2 (Mulmuley & Sohoni 2008).[12] *A* multiplicity obstruction *for the lower bound $\mathcal{C}_{hard} \not\subseteq \mathcal{C}_{easy}$ is an irreducible equivalence class λ such that $m_\lambda(\overline{\mathcal{C}_{easy}}) < m_\lambda(\overline{\mathcal{C}_{hard}})$. An* occurrence obstruction *or* geometric obstruction *for $\mathcal{C}_{hard} \not\subseteq \mathcal{C}_{easy}$ is a multiplicity obstruction which further has $m_\lambda(\overline{\mathcal{C}_{easy}}) = 0$; that is, every test module in the equivalence class λ vanishes everywhere on $\mathcal{C}_{easy}$.*

The existence of a multiplicity obstruction λ implies the existence of a separating module: The number of test modules of type λ that vanish on $\overline{\mathcal{C}}$ and the number of test modules of type λ that are supported on $\overline{\mathcal{C}}$ must add up to m_λ, hence $m_\lambda(\overline{\mathcal{C}_{easy}}) < m_\lambda(\overline{\mathcal{C}_{hard}})$

[12] Although only geometric obstructions were explicitly defined in Mulmuley & Sohoni (2008), multiplicity obstructions were essentially defined there: See the sentence just before Mulmuley & Sohoni (2008, Definition 1.2).

if and only if there are more test modules of type λ that vanish on $\overline{\mathcal{C}_{easy}}$ than vanish on $\overline{\mathcal{C}_{hard}}$. Hence, there must be some test module of type λ that vanishes on $\overline{\mathcal{C}_{easy}}$ but not on $\overline{\mathcal{C}_{hard}}$, i.e., a separating module. These are referred to as "obstructions" because they obstruct the inclusion $\mathcal{C}_{hard} \subseteq \mathcal{C}_{easy}$, much as a K_5-minor obstructs a planar embedding of a graph.

One advantage of considering multiplicities instead of test modules is that it opens the possibility of using purely representation-theoretic techniques to understand the multiplicities, as is being pursued in GCT (e.g., Adsul, Sohoni & Subrahmanyam 2009; Blasiak 2012; Blasiak, Mulmuley & Sohoni 2015; Bürgisser, Christandl & Ikenmeyer 2011a). To see how this is possible—that is, how one can discuss multiplicity obstructions without reference to actual test polynomials or modules thereof—we must mention a bit more about the representation theory of GL_n and S_n. Over $\mathbb{C}$, the irreducible representations of these groups have been classified for over 100 years (see, e.g., Fulton & Harris 1991).[13] The equivalence classes of irreducible representations[13] are in bijective correspondence with integer partitions—partitions with at most n parts in the case of $\mathrm{GL}_n(\mathbb{C})$, and partitions of the number n in the case of S_n. The use of partitions enables us to talk about the multiplicities m_λ and $m_\lambda(\overline{\mathcal{C}})$ without reference to any particular (test) module. This is just one of the advantages of the representation-theoretic viewpoint; we discuss two other advantages below.

However, we note that computing the values $m_\lambda(\overline{\mathcal{C}})$ seems to be incredibly difficult, not only in the complexity-theoretic sense (they are believed to be #P-hard to compute), but also in the practical sense. There are many #P-hard representation-theoretic multiplicities that can nonetheless be computed efficiently in small instances (e.g., Bürgisser & Ikenmeyer 2009; Bürgisser & Ikenmeyer 2013; Pak & Panova 2014), but even for small instances computing $m_\lambda(\overline{\mathcal{C}}_n)$ seems to be difficult. This should not be particularly surprising, as knowing more about these multiplicities potentially gives a lot of information about lower bounds. What

[13] In the case of $\mathrm{GL}_n(\mathbb{C})$, this only applies to *polynomial* irreducible representations, namely those representations $\varphi\colon \mathrm{GL}_n(\mathbb{C}) \to \mathrm{GL}_m(\mathbb{C})$ where each entry of $\varphi(X)$ is a polynomial in the entries of $X \in \mathrm{GL}_n(\mathbb{C})$.

little is known is summarized in the thesis Ikenmeyer (2012) and the survey Landsberg (2014a).

4.2. Understanding old lower bounds better (even tight ones!) In this paper, we show that most algebraic circuit lower bounds yield separating modules, but typically just one separating module for each lower bound. While this suffices for the lower bound, considering other separating modules that can be used for a given lower bound (or non-separating test modules) may give deeper insight. Indeed, by Fact 2.8, this is equivalent to knowing which other invariant properties defined by polynomials can be used (or not) for a lower bound. Understanding which (invariant) properties a complexity class has is surely a task worth undertaking, even for lower bounds that are already tight or as good as we want.

However, trying to understand all such test modules is an enormous task. It is not just analogous to asking for new proofs of old lower bounds—for example, just asking for a single new separating module for the lower bound—but rather is analogous to understanding *all* possible proofs of a given lower bound. Instead, the difference between separating modules and multiplicity obstructions suggests a more feasible step in this direction which may well be within reach:

Open Question 4.3. *Can the proofs of lower bounds mentioned in this paper be upgraded from separating modules to multiplicity (or stronger: occurrence) obstructions? A first step—potentially useful regardless of the answer—is to determine the labels (partitions, see* Section 4.1*) of the separating modules in the lower bounds mentioned in this paper.*

4.3. Boolean circuit lower bounds.

Observation 4.4. *There is a separating S_n-module for any Boolean circuit lower bound against a permutation-invariant complexity class (which includes all natural classes, see* Observation 3.2*).*

Proof. By Fact A.1 for S_n, we only need to argue that the complexity class is defined by test polynomials. As the space of

Boolean functions on n variables is finite, every property of n-variable Boolean functions is finite and hence defined by test polynomials over $\mathbb{F}_2$. □

Despite the fact that Observation 4.4 says that separating modules can be used without loss of generality for Boolean circuit lower bounds, we find this observation alone somewhat unsatisfying. However, as with the results of Razborov, Smolensky, and Grigoriev–Karpinski over finite fields (see Section 5.4), we believe that many Boolean circuit lower bounds in fact yield separating modules in a very *direct and natural* manner.

Although we have not yet verified this for many known Boolean lower bounds, we explain why we expect this to be the case. By the discussion in Appendix B.3, it is reasonable to expect that lower bounds use properties Π which are *naturally* defined by some logical combination of the vanishing of some polynomials and the non-vanishing of other polynomials. The preceding observation tells us that the properties used *can* be defined by the vanishing of some test polynomials; the key here is the *naturality* (in the usual sense of the word, not the Razborov–Rudich sense). In such generality, we will argue for their naturality based on the number of polynomials used and their degree.

Observation 4.5. *Let Π be a subset of the vector space $\mathbb{F}_q^n$, defined as $\Pi = C(\Pi_1, \ldots, \Pi_k)$ where each Π_i is defined by the vanishing of a set of n-variable polynomials, and C is a logical combination (using union, intersection, and complement). Let $N \subseteq \{1, \ldots, k\}$ be the set of indices i such that Π_i appears complemented in C an odd number of times. Let m_i denote the number of polynomials used to define Π_i, and let d_{ij} denote the degree of the j-th polynomial used to define Π_i.*

Then Π can be defined by the vanishing of a set of at most $|N| + \sum_{i \notin N} m_i$ polynomials, each of degree at most

$$(q-1)\min\{n, \max_i\{\sum_{j=1}^{m_i} d_{ij}\}\}.$$

Proof. For simplicity, we start with the case $q = 2$, and the general case is similar. Putting the logical combination C into disjunctive normal form, Π can be naturally expressed as a union of properties of the form $\Pi_i \backslash \Pi'_i = \Pi_i \cap \Pi'^c_i$, where we have re-indexed the Π_i if necessary, and Π'^c_i denotes the complement of Π'_i. Say Π'_i is defined by the vanishing of the polynomials $f_1(x_1, \dots, x_n) = \cdots = f_{m_{i'}}(\mathbf{x}) = 0$. Then its complement is most naturally defined by the *non*-vanishing of at least one of the f_i. However, the complement Π'^c_i can also be defined by the *vanishing* of the single polynomial $\prod_{i=1}^k (f_i(\mathbf{x}) - 1)$. Furthermore, by applying $x_i^2 = x_i$, we may take the degree of this single polynomial to be at most $\min\{n, \sum_{i=1}^k \deg(f_i)\}$. A similar idea works over any finite field $\mathbb{F}_q$: use $\prod_{0 \neq \alpha \in \mathbb{F}_q} (f_i(\mathbf{x}) - \alpha)$ in place of $f_i(\mathbf{x}) - 1$, reduce by $x_i^q = x_i$, and the resulting degree is at most $(q-1)\min\{n, \sum_i \deg(f_i)\}$. □

In terms of constructivity, we thus do not lose much by considering Π'^c_i as being defined by the vanishing rather than non-vanishing of test polynomials: The single polynomial defining Π'^c_i has low degree, and there is only one such polynomial, so the number of polynomials used to define the property also does not increase.

One might argue that using $\prod(f_i(\mathbf{x}) - 1) = 0$ rather than the non-vanishing of some f_i is unnatural or violates the spirit of the lower bound proof that used property Π. However, if this were really the case, then the lower bound proof would hold for the vanishing/non-vanishing of some f_i as formal polynomials, and hence would work over fields larger than $\mathbb{F}_2$ and, in particular, would hold over the algebraic closure $\overline{\mathbb{F}}_2$. We are not aware of Boolean lower bounds that extend to any infinite fields. In this sense, the use of finiteness in Observation 4.4 seems less of a kludge to us and more an essential feature of the current techniques for Boolean circuit lower bounds.

4.4. Other lower bounds? Although we have obviously not considered *all* known lower bounds, we have considered a wide cross section of them in this paper. Of the lower bounds which we actively tried to fit into this framework but have not yet been able to do so, most use heavily machine-based diagonalization: for example, the (non)deterministic time and space hierarchies (Cook 1973; Hartmanis & Stearns 1965), uniform lower bounds on the permanent (Allender 1999; Allender & Gore 1994; Koiran & Perifel 2009), time-space trade-offs for SAT (Buss & Williams 2012; Diehl, van Melkebeek & Williams 2011; Fortnow 2000; Fortnow, Lipton, van Melkebeek & Viglas 2005; Williams 2006, 2008), $\mathsf{\Sigma_2 P} \cap \mathsf{\Pi_2 P} \not\subseteq \mathsf{SIZE}(n^k)$ (Kannan 1982), and the related result $\mathsf{MA_{EXP}} \not\subseteq \mathsf{P/poly}$ (Buhrman, Fortnow & Thierauf 1998).

Remark 4.6. *Although from one viewpoint Kannan's result rests crucially on the non-uniform circuit-size hierarchy, for the purposes of this discussion the key fact he shows is that a* uniform $\mathsf{\Sigma_4 P}$*-machine is powerful enough to use the circuit-size hierarchy to diagonalize against* $\mathsf{SIZE}(n^k)$*. The same remark applies to the result* $\mathsf{MA_{EXP}} \not\subseteq \mathsf{P/poly}$*, as it uses Kannan's result in an essential way.*

The recent lower bound $\mathsf{NEXP} \not\subseteq \mathsf{ACC}^0$ (Williams 2014) provides an interesting crucible. It is a non-uniform lower bound against a permutation-invariant Boolean complexity class; hence by Observation 4.4, there *exists* a separating S_n-module proving $\mathsf{NEXP} \not\subseteq \mathsf{ACC}^0$. However, the proof uses the non-deterministic time hierarchy in a seemingly crucial way. Extracting a *natural* separating module from Williams's proof may be a first step toward extending the representation-theoretic framework to include uniform lower bounds.

One very interesting technique which we have not yet been able to fit into the representation-theoretic framework and which is only partially uniform comes from Jansen & Santhanam (2012, 2013). The key property they use is the existence of $\mathbb{Z}$ hitting sets whose bit descriptions can be encoded by small uniform (or at least succinct) circuits. This combination of algebraic (hitting sets) and Boolean (bit descriptions) frameworks in the same breath makes it difficult to even formulate their proofs in a single algebraic setting, let alone translate them into separating modules.

Finally, the standard counting argument of Shannon (1949) also seems difficult to put into this representation-theoretic framework. By Observation 4.4, there exists a separating S_n-module for this lower bound. However, finding a natural separating module seems difficult, as Shannon counts the functions in $\mathcal{C}_{easy}$ (in this case, $\mathsf{SIZE}(2^n/n)$), rather than using some property shared by these functions. Such counting arguments are also likely to be unnatural in the sense of Razborov & Rudich (1997, Section 2.1). But this is not a problem for either them or us, as such counting arguments have yet to prove lower bounds on any explicit function.

4.5. Finite fields and positive characteristic. There is a mismatch between the current lower bounds over finite fields and standard techniques in algebraic geometry. The issue is that all the current lower bounds over finite fields that we are aware of depend crucially not just on positive characteristic, but on the size of the field. This means that none of the current lower bounds over finite fields extend to the algebraic closure $\overline{\mathbb{F}}_p$. This is in contrast to the usual approach to finite fields in algebraic geometry, which is (roughly) to first work over their algebraic closures $\overline{\mathbb{F}}_q$—where algebraic geometry and representation theory are nicer—and then to pass to the $\mathbb{F}_q$ points. The $\mathbb{F}_q$-points can be recovered from $\overline{\mathbb{F}}_q$ as the fixed points of the Frobenius map $x \mapsto x^q$, just as $\mathbb{R}$ points can be recovered from $\mathbb{C}$ as the fixed points of the complex conjugation map. The dynamics of the Frobenius map are often very useful. In particular, over $\overline{\mathbb{F}}_q$ Hilbert's Nullstellensatz holds and every matrix admits an eigenvector. This process is exactly analogous to (but more complicated than) considering complex solutions, eigenvectors, etc., in order to study equations, matrices, etc., over $\mathbb{R}$.

As we already mentioned, even if the characteristic is held constant but the field size is allowed to grow at a modest pace with the size of the input, the current lower bounds seem to disappear completely. The essential issue here seems to be that the method of approximations is typically used to "throw away" points which are in the *complement* of an algebraic set. Over finite fields, one then argues that these "erroneous points" are not too numerous, but over any infinite field, *almost all* points will be "erroneous," as an algebraic set has dimension strictly smaller than that of the ambient space.

It thus seems to us that the limits of our knowledge are not so much in finding lower bounds for depth 3 algebraic circuits in characteristic zero, as is often stated, but for finding lower bounds for depth 3 algebraic circuits over any given infinite field, including $\overline{\mathbb{F}}_p$. The chasm at depth 4 (Agrawal & Vinay 2008; Koiran 2012; Tavenas 2013) holds over an arbitrary field, but these observations lead us to wonder:

Open Question 4.7. *Is there a chasm at depth 3 over the algebraically closed field $\overline{\mathbb{F}}_p$ for any constant prime $p > 0$? More positively, can we prove depth 3 lower bounds over $\overline{\mathbb{F}}_p$ without proving lower bounds on arbitrary algebraic circuits?*

The current proofs of the chasm at depth 3 (Gupta, Kamath, Kayal & Saptharishi 2013; Tavenas 2013) only seem to work in characteristic zero or over a field of (growing) characteristic greater than the degree d of the polynomial, as they use a trick of Fischer (1994) which requires dividing by $2^{d-1}d!$. On the other hand, taking the positive route leads one to consider what properties of characteristic p might be leveraged without appealing to finiteness, which seems like a potentially useful exercise.

4.6. Explicitness and constructivity. Mulmuley (2010) and Williams (2013) have both argued—based on separate but related ideas—for the necessity of constructive methods in proving lower bounds. We can use the representation-theoretic viewpoint to quantify the explicitness or constructivity of known proofs in various ways.

One measure of constructivity is the degree and number (dimension) of test polynomials used. As in the context of Razborov & Rudich (1997) and Williams (2013), we should expect to measure this degree as a function of something like the size of the truth table of the input polynomials involved. In an algebraic context, we might replace truth table size by the number of monomials. For polynomials of degree $O(n)$ in poly(n) variables, the number of monomials is $2^{O(n \log n)}$, which is comparable to truth table size.

Another more delicate measure of constructivity is the complexity of verifying that a given test module is indeed a separating module. This is related to our discussion in Appendix B.3.

Using the fact that partitions classify the irreducible representation of GL_n or S_n over $\mathbb{C}$, we get another measure of constructivity. In general, the dimension of an irreducible representation can be exponential in the bit-size of its corresponding partition, so the partition can serve as a succinct label of an equivalence class of representations. One can then consider the computational complexity of constructing from 0^n a partition corresponding to a multiplicity (or occurrence) obstruction for a non-uniform lower bound at input length n. Mulmuley (2010) conjectures that this construction problem can be solved in P for occurrence obstructions in the context of permanent versus determinant and NP versus $\mathsf{P/poly}$. In fact, Mulmuley suggests that finding a polynomial-time algorithm to verify whether a given λ_n is the label of an obstruction is a crucial first step toward proving the existence of obstructions unconditionally. This suggests a strengthening of Open Question 4.3:

Open Question 4.8. *For the lower bounds mentioned in this paper, are there multiplicity obstructions for which the label λ_n of the obstruction at input length n can be computed in* $\mathrm{poly}(n)$*-time?*

In Open Question 4.3, we suggest a first step that would provide natural candidates for labels λ that might be multiplicity obstructions. However, we caution that although some occurrence obstructions are known in the context of matrix multiplication (Bürgisser & Ikenmeyer 2013), there are other separating modules for matrix multiplication which are either suspected or known not to be multiplicity obstructions (Hauenstein, Ikenmeyer & Landsberg 2013; Landsberg & Ottaviani 2011) (because the relevant multiplicities have been computed and they do not satisfy the appropriate inequality). That is, we do not necessarily expect that the known separating modules are all in fact multiplicity obstructions, but they are certainly first candidates to check (as in Open Question 4.3).

The more general question of verification is also interesting:

Open Question 4.9. *For any lower bound mentioned here, what is the complexity of verifying multiplicity or occurrence obstruc-*

tions? That is, given λ_n, what is the complexity of verifying that λ_n is indeed a multiplicity obstruction?

5. Most algebraic circuit lower bounds yield separating modules

In this section, we show how all of the bounds listed in the introduction give separating modules. Rather than recalling all of these proofs and stating a separate proposition for the existence of a separating module for each of these bounds (as in Section 2), we use a more concise format. Furthermore, we have not included all the results from every paper we consider, but only a representative result from each paper (or sometimes, from each technique). We nonetheless expect that the other results in these papers and using these techniques also yield separating modules.

5.1. Methods based on partial derivatives. Of all the lower bounds we discuss, those based on partial derivatives and their variants (shifted partial derivatives, shifted partial derivatives of bounded support, etc.) are the easiest to extract separating modules from, so we begin with those.

Nisan–Wigderson partial derivatives
Hard function: Elementary symmetric function $e_{n/4,n}$
Complexity class: Homogeneous depth 3 algebraic circuits in characteristic zero
Lower bound: Size $2^{\Omega(n)}$ (Nisan & Wigderson 1996/97)
Invariance: $\mathbb{F}$-linear ($\mathrm{GL}_n(\mathbb{F})$), characteristic zero
Separating module: The $(r+1) \times (r+1)$ minors of the partial derivative matrix M_f, as in the proof of Proposition 2.12. □

Permanent and determinant versus depth 4
Hard function: perm_n or $\det_n$
Complexity class: Depth 4 $\Sigma\Pi\Sigma\Pi$ algebraic circuits with bottom fan-in $O(\sqrt{n})$
Lower bound: Size $2^{\Omega(\sqrt{n})}$ (Gupta, Kamath, Kayal & Saptharishi 2012)
Invariance: $\mathbb{C}$-linear ($\mathrm{GL}_{n^2}(\mathbb{C})$)

Separating module: The outline of the proof of this lower bound is very similar to that for the Nisan–Wigderson lower bound above. However, the key property used here is slightly more complicated. Rather than considering the dimension of the space of all partial derivatives $\partial(f)$, they consider the dimension of the space of *shifted* partial derivatives, which are products of polynomials of some degree ℓ with the partial derivatives of f. Following their notation, we write $\partial^{=k}(f)_{\leq \ell}$ for the space of k-th order partial derivatives multiplied by polynomials of degree $\leq \ell$. As in the above case, we build a matrix $\tilde{M}_f$ whose rank is exactly the dimension of $\partial^{=k}(f)_{\leq \ell}$, and then the $r \times r$ minors of this matrix provide the separating module, for appropriately specified r, k, and ℓ.

As above, the columns of $\tilde{M}_f$ will be indexed by monomials $\mathbf{x}^e$, and the rows will be indexed by pairs $(\mathbf{x}^d, \partial^c)$ of a monomial and a partial derivative operator. (Here $c \in \mathbb{Z}^n_{\geq 0}$, and ∂^c denotes $\partial / \partial x_1^{c_1} \cdots \partial x_n^{c_n}$.) Then we proceed as in the above case.

Fact 5.1. *The disjunction (union) of two invariant properties defined by test polynomials is again an invariant property defined by test polynomials.*

Proof. Let V, W be test modules. First one verifies that the product $V \cdot W \overset{def}{=} \{\sum_i f_i g_i | f_i \in V, g_i \in W\}$ is a test module. Then $V \cdot W$ defines the union of the properties defined by V and W. For let f be an input polynomial. If every test polynomial $t \in V$ vanishes at f, then so does every test polynomial in $V \cdot W$. Similarly for W. Conversely, if some test polynomial $t_1 \in V$ does not vanish at f, and some test polynomial $t_2 \in W$ does not vanish at f, then $t_1 t_2 \in V \cdot W$ does not vanish at f. □

Multilinear formulas

Hard function: $\det_n$ or perm_n

Complexity class: (Syntactic) multilinear formulas in characteristic zero

Lower bound: Size $\Omega(n^{\log n})$ (Raz 2009)

Invariance: Permutation (S_n)

Separating module: Raz combines the above ideas on $\dim \partial(f)$ with random restrictions, making the separating module here a little more complicated than in the above examples. Raz explicitly defines a matrix of partial derivatives, similar to that in the above two examples, which he also denotes M_f. The random restrictions used in Raz (2009, Section 5) take the form $\rho(x_i, x_j, x_k, x_\ell) = (1, 1, y_m, z_m)$, where the i, j, k, ℓ used are of a particular form, and the image may be re-ordered in one of the two possible ways. In particular, for each input length n there are only finitely many such restrictions to consider.

He then shows a lower bound on $\operatorname{rk} M_{\det(\rho(X))}$ and $\operatorname{rk} M_{\mathrm{perm}(\rho(X))}$ under *any* such restriction ρ, and using a probabilistic argument shows that there *exists* a restriction making $\operatorname{rk} M_{f(\rho(X))}$ small when f is computed by a multilinear formula of size $n^{o(\log n)}$. Hence, the property he is using is that there *exists* a restriction ρ as in his Section 5 that makes $\operatorname{rk} M_{f(\rho(X))} \leq r$ for appropriately chosen r.

For a given restriction ρ, we get a test S_n-module V_ρ consisting of the $(r + 1) \times (r + 1)$ minors of $M_{f(\rho(X))}$. This test S_n-module vanishes if and only if $\operatorname{rk} M_{f(\rho(X))} \leq r$. The separating module is then the product over all (finitely many) ρ of the V_ρ (use Fact 5.1 and induction). □

Remark 5.2. *Although bounding the rank of a matrix of partial derivatives is linear-invariant, the property of being multilinear is* not *linear-invariant, though it is permutation-invariant. Hence, despite using a bound on the dimension of partial derivatives, it was to be expected that at some point in the proof a property would be used that was only permutation-invariant and not linear-invariant. Although Raz uses multilinearity elsewhere in his proof, even in the brief outline above we see that the type of random restrictions used is only permutation-invariant and not linear-invariant.*

5.2. Elusive functions. Raz (2010a) defined the notion of elusive function, which we recall briefly here, because its relationship with separating modules is a bit more subtle than the other results we cover. An (π, δ)-elusive function over a field $\mathbb{F}$ is a polynomial function $f\colon \mathbb{F}^n \to \mathbb{F}^m$—that is, a collection of m polynomials $f_1, \ldots, f_m\colon \mathbb{F}^n \to \mathbb{F}$—such that the image of f is not contained in the image of any polynomial map $\Gamma\colon \mathbb{F}^\pi \to \mathbb{F}^m$ of total degree at most δ. ("π" for "number of **p**arameters" and "δ" for "**d**egree.")

Using the fact that the set of easily computable functions is the image of a single low-degree map Γ_G (Raz 2010a, Proposition 3.3), Raz showed that sufficiently explicit elusive functions imply lower bounds on accordingly explicit functions (e.g., in VNP).

Note that, already from what we have said, to prove a lower bound it suffices to prove that the image of f is not contained in the single map Γ_G of Raz's Proposition 3.3. This idea essentially goes back at least to Heintz & Sieveking (1980), and similar ideas are used explicitly by Mulmuley & Sohoni (2001, 2008) and implicitly or explicitly whenever border rank is invoked in the context of matrix multiplication (e.g., Bini 1980; Bürgisser & Ikenmeyer 2013; Coppersmith & Winograd 1990; Schönhage 1981; Stothers 2010; Strassen 1983; Vassilevska Williams 2012). I believe the hope in using the much stronger notion of elusivity is that it may make it easier to prove lower bounds by in fact trying to prove a stronger statement (such counterintuitive trade-offs are not infrequent throughout mathematics).

A priori, a lower bound proof using elusive functions need not use test polynomials. However, all known lower bounds to date that have been proven using elusive functions (Lê 2010, 2013; Raz 2010a)—and even the intriguing suggestions of Vân Lê on how to prove further lower bounds using elusive functions—do in fact use test polynomials and fit into the framework of this paper. We use Raz's result as an example.

Elusive functions versus bounded-depth circuits
Hard function: $\sum_{a,b\in[n]} w_a z_b \prod_{i\in[(\log_2 n)/20]} x_{i,a+ib}$ (a function with a depth 2 formula of size $O(n^2 \log n)$)
Complexity class: Depth d algebraic circuits over an arbitrary field $\mathbb{F}$
Lower bound: Size $\geq n^{1+\Omega(1/d)}$ (Raz 2010a)
Invariance: $\mathbb{F}$-linear ($\mathrm{GL}_n(\mathbb{F})$), $\mathbb{F}$ arbitrary
Separating module: The useful property here is somewhat harder to describe explicitly, because for each polynomial mapping Γ from $\mathbb{F}^\pi$ to $\mathbb{F}^m$ of degree δ, Raz shows the existence of a test polynomial that vanishes on $\mathrm{Im}(f)$ but not on $\mathrm{Im}(\Gamma)$ by a dimension-counting argument (Raz 2010a, the last few paragraphs before Section 4.2). As mentioned above, to prove the lower bound it suffices to show

that $\mathrm{Im}(f)$ is not contained in $\mathrm{Im}(\Gamma_G)$, where Γ_G is the function constructed in Raz's Proposition 3.3. The function Γ_G essentially takes as input the coefficients on an algebraic circuit in what Raz calls "homogeneous normal form" (Raz 2010a, Definition 2.2). As homogeneous normal form includes that the bottom layer of gates are linear combination gates, the image of Γ_G is $\mathrm{GL}_n(\mathbb{F})$-invariant. Hence, we may take the test polynomial corresponding to Γ_G and consider the smallest separating GL_n-module containing it. □

Vân Lê extended Raz's work on elusive functions in several ways, in particular, by introducing the notion of *weakly* elusive function. This has the same definition as elusive function, except instead of not being contained in the image of *any* polynomial mapping $\Gamma\colon \mathbb{F}^\pi \to \mathbb{F}^m$ of degree δ, weak elusivity only requires not being contained in the image of any *homogeneous* such polynomial mapping Γ. Given that the map Γ_G defining the collection of functions computed by a circuit with graph G is homogeneous, this is certainly enough for lower bounds. As the map Γ_G is furthermore GL_n-equivariant, we suggest taking this one step further in the following definition. Recall that a map $f\colon X \to Y$ between two sets with actions of G on them is called *G-equivariant* if it commutes with the action of G, i.e., for any $g \in G$ and any $x \in X$, $f(g(x)) = g(f(x))$.

Definition 5.3 (G-elusive). *Let G be a group with an algebraic action on $\mathbb{F}^m$. A polynomial function $\mathbb{F}^n \to \mathbb{F}^m$ is (π, δ) G-*elusive *if its image is not contained in the image of any G-equivariant polynomial map $\mathbb{F}^\pi \to \mathbb{F}^m$ of degree at most δ.*

Observation 5.4. *The existence of explicit G-elusive functions implies algebraic circuit lower bounds, for the same notions of explicit and bounds considered by* Raz (2010a) *and* Lê (2013).

We hope that this maintains the spirit of elusive functions in going for something that is stronger than necessary but in a useful way, yet may be easier to achieve than looking for fully elusive functions.

Open Question 5.5. *Prove new lower bounds using G-elusivity.*

5.3. Methods using (semi-)algebraic varieties. For methods such as the degree bound (Baur & Strassen 1983; Strassen 1972/73) and the connected components technique (Ben-Or 1983), the most natural input objects to use are (semi-)algebraic varieties themselves. In other words, we need to replace the input space $\mathrm{Poly}^d(\mathbf{x})$ with a space whose points correspond to varieties. Such spaces have been constructed in (semi-)algebraic geometry, but their construction is not as elementary as in the above results. In both cases, the basic idea is that the input objects will in fact be systems of equations (which, in turn, define algebraic sets), and the test variables are then the coefficients of these systems of equations.

Surprisingly, the use of these "parameter spaces of algebraic sets" makes putting these results into the representation-theoretic viewpoint technically more complicated than the above results, despite the fact that these bounds were discovered considerably earlier.[14]

The degree bound

Hard function: Computing all elementary symmetric functions $e_{1,n}, \ldots, e_{n,n}$ together
Complexity class: Algebraic circuits over an infinite field
Lower bound: Size $\Omega(n \log n)$ (Strassen 1972/73)
Invariance: $\mathbb{F}$-affine ($\mathrm{AGL}_n(\mathbb{F})$), $\mathbb{F}$ infinite
Separating module: The key property used here is the degree of a projective algebraic set. Although the degree has a nice geometric definition (in characteristic zero), here we recall the algebraic definition as it lends itself more readily to the definition of the separating module. Let V be an algebraic subset of the projective space $\mathbb{P}(\mathbb{F}^n)$, and let $I \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be the homogeneous ideal of all polynomials that vanish on V. In particular, I can be written as the direct sum $\bigoplus_d I_d$ of its homogeneous subsets I_d, which consist of those polynomials in I of degree exactly d. The *Hilbert function*

[14] Griesser (1986) showed that the degree bound applies in the setting of approximative complexity; however, his proof uses the usual degree bound as a black box and does not reveal that the property of degree is defined by test polynomials, let alone defined by a separating module.

of I is then $h_I(d) \overset{def}{=} \dim_{\mathbb{F}} I_d$. Hilbert showed (see, e.g., Cox, Little & O'Shea 1997, Section 9.3 or Eisenbud 1995, Theorem 1.11) that for all sufficiently large d, $h_I(d)$ agrees with a polynomial $p_I(d)$, known as the *Hilbert polynomial* of I or V. The *degree* of V is then essentially the leading coefficient of the Hilbert polynomial $p_I(d)$.[15]

For the input space, we may use either the Chow variety (see, e.g., Danilov 1994, Chapter 3, Section 7) or the Hilbert scheme (see, e.g., Grothendieck 1995). The Chow variety is essentially the "space of projective algebraic sets," and the Hilbert scheme is essentially the "space of homogeneous ideals in $\mathbb{F}[x_1, \ldots, x_n]$." The Chow variety is in fact a disjoint union over pairs (d, D) of the variety of projective algebraic sets of degree d and dimension D. Similarly, the Hilbert scheme is the disjoint union over Hilbert polynomials $p(\cdot)$ of the scheme of homogeneous ideals with Hilbert polynomial $p_I = p$. In either case, showing that two varieties have different degrees then amounts to showing that these varieties, as points in the space of varieties, live in different connected components of the Chow variety or Hilbert scheme.

Finally, being in a given component of a variety (or scheme) is defined by the vanishing of some (test) polynomials. As the Hilbert polynomial, and in particular the degree and dimension, is an affine invariant of a projective algebraic set, the components of the Chow variety and Hilbert scheme are also affine-invariant. Hence, by Fact A.1 for affine invariance, there is a separating module. □

Remark 5.6. *We admit that, although technically the preceding shows that Strassen's degree bound can be phrased in terms of separating modules, this is a somewhat odd way of viewing it, and coming up with the degree bound from this viewpoint would be difficult. However, in this case there is a slightly different viewpoint that unifies this result with several others such as lower bounds on low-depth circuits and with the GCT Program*

[15] More precisely, the degree of V is the leading coefficient of its Hilbert polynomial divided by the factorial of the degree of the Hilbert polynomial. In an unfortunate twist of terminological fate, it turns out that the dimension of V in the usual sense is equal to the degree of its Hilbert polynomial.

in general. Namely, the degree bound (Strassen 1972/73), Nisan & Wigderson (1996/97)*, and* Gupta, Kamath, Kayal & Saptharishi (2012) *can be viewed as using information about the Hilbert function of a variety and its singular loci, as we now explain. As already mentioned, the degree of a variety, as used by* Strassen (1972/73)*, is determined by the leading coefficient of the Hilbert polynomial. Given a polynomial f, the set of points at which all of the k-th partials of f vanish is the same as the set of points at which f vanishes with multiplicity at least $k+1$. The dimension of the space of k-th partials is then partial information about the Hilbert function—namely, its value in the least degree in which it is nonzero—of the variety of points where f vanishes to multiplicity at least $k+1$. This applies to both the homogeneous depth 3* (Nisan & Wigderson 1996/97) *and depth 4* (Gupta, Kamath, Kayal & Saptharishi 2012) *lower bounds. Considering* shifted partials *of f then amounts to considering the Hilbert functions at higher degrees; this is also explained in the original paper* (Gupta, Kamath, Kayal & Saptharishi 2012, just before Section 3).

One can also consider a so-called G-Hilbert scheme, which parametrizes all G-invariant subvarieties of a given variety (Brion 2011)*. Here, rather than getting one component for each Hilbert polynomial, one gets a part of the G-Hilbert scheme for each "G-Hilbert function," which is just the multiplicity function of G-modules, namely the $m_\lambda(V)$ we considered in* Section 4.1*. That is, the G-Hilbert function of a G-invariant variety V gives the multiplicity of each irreducible G-module that vanishes on V. It is precisely these G-Hilbert functions that are the focus of study in the GCT Program* (Mulmuley 2011a; Mulmuley & Sohoni 2001, 2008) *(see* Section 4.1 *for an overview of how this relates to separating modules).*

The following two results are in the model of real semi-algebraic decision trees. Here we do not yet know of a good analog of separating module—though we hope to make that the subject of future work—but we can still show that the properties used in the lower bounds are invariant under a non-trivial group action and defined by equalities and inequalities between (test) polynomials. In Remark 5.8, after we discuss the proof for the seminal result of

Ben-Or (1983), we show that the use of inequalities is necessary for the properties used in these particular lower bounds.

The reason we do not yet have a good notion of separating module in the semi-algebraic setting is closely related to the fact that there is not really a good semi-algebraic analog of the coordinate ring or defining ideal of an algebraic variety. For *basic* invariant semi-algebraic sets—of the form $\{x : f_1(x) = \cdots = f_k(x) = 0, g_1(x) > 0, g_2(x) > 0, \ldots, g_\ell(x) > 0\}$—we can give a fairly good analog of separating module using standard tools from real algebraic geometry (e.g., Bochnak, Coste & Roy 1998, Chapter 4), but general semi-algebraic sets are unions of these basic sets (the real analog of constructible sets, see Definition B.4), and cannot always be written as a single basic semi-algebraic set.

Algebraic decision trees for sorting

Hard function: Element distinctness (note that element distinctness reduces to sorting)

Complexity class: Real semi-algebraic decision trees

Lower bound: Depth $\Omega(n \log n)$ (Ben-Or 1983)

Invariance: $\mathbb{R}$-affine ($\mathrm{AGL}_n(\mathbb{R})$), and more generally any topological homeomorphism of $\mathbb{R}^n$

Invariant property defined by polynomial inequalities: The key property used here is the number of connected components of a semi-algebraic variety—that is, a subset of $\mathbb{R}^n$ defined by a collection of polynomial equalities and inequalities. The number of connected components is clearly affine-invariant; we recall here how Hardt's Triviality Theorem implies that it is in fact defined by a collection of test polynomial equalities and inequalities. The use of inequalities here is unavoidable: See Remark 5.8 below.

A special case of Hardt's Triviality Theorem (Hardt 1980) (see, e.g., Basu, Pollack & Roy 2006, Section 5.8 for a textbook treatment) says that for any continuous semi-algebraic map $\pi\colon S \to \mathbb{R}^N$ from a semi-algebraic set $S \subseteq \mathbb{R}^n$, there is a finite partition of $\mathbb{R}^N$ into semi-algebraic sets $\mathbb{R}^N = \bigcup_{i=1}^k T_i$ such that for each i and every $x \in T_i$, $T_i \times \pi^{-1}(x)$ is semi-algebraically homeomorphic to $\pi^{-1}(T_i)$. In particular, this implies that for each i, if $x, y \in T_i$ then $\pi^{-1}(x)$ and $\pi^{-1}(y)$ have the same number of connected components.

Now, consider a collection of polynomial equalities and inequalities of degree $\leq d$ in n variables $x_1, \ldots, x_n$:

$$\begin{array}{ll} \sum_e a_{1,e}\mathbf{x}^e + a_1 = 0, & \ldots, \sum_e a_{m,e}\mathbf{x}^e + a_m = 0 \\ \sum_e a_{m+1,e}\mathbf{x}^e + a_{m+1} \geq 0, & \ldots, \sum_e a_{m+s,e}\mathbf{x}^e + a_{m+s} \geq 0 \\ \sum_e a_{m+s+1,e}\mathbf{x}^e + a_{m+s+1} > 0, & \ldots, \sum_e a_{h,e}\mathbf{x}^e + a_h > 0 \end{array} \tag{5.7}$$

We may consider the $a_{i,e}$s and a_is as variables rather than constants; suppose in total there are N such variables. Then the x_is are coordinates on $\mathbb{R}^n$ and the $a_{i,e}$s are coordinates on $\mathbb{R}^N$. Equations (5.7) thus define a semi-algebraic subset $S \subseteq \mathbb{R}^n \times \mathbb{R}^N$. Let $\overline{\pi}\colon \mathbb{R}^n \times \mathbb{R}^N \to \mathbb{R}^N$ be the projection onto the second factor, and let $\pi\colon S \to \mathbb{R}^N$ be the restriction of $\overline{\pi}$ to S. For any given numerical values $\mathbf{a} \in \mathbb{R}^N$, let $V_{\mathbf{a}} \subseteq \mathbb{R}^n$ denote the semi-algebraic subset defined by (5.7). Then, $\pi^{-1}(\mathbf{a}) = V_{\mathbf{a}} \times \{\mathbf{a}\} \cong V_{\mathbf{a}}$ (where $\cong$ here denotes semi-algebraic homeomorphism).

Finally, by Hardt's Triviality Theorem, there is a semi-algebraic partition $\mathbb{R}^N = \bigcup_{i=1}^k T_i$ such that for any $\mathbf{a}$ and $\mathbf{a}'$ in the same T_i, $V_{\mathbf{a}}$ and $V_{\mathbf{a}'}$ have the same number of connected components. Hence, the collection of equations of the form (5.7) that define a semi-algebraic variety with c connected components is the semi-algebraic set $\bigcup\{T_i | \pi^{-1}(\mathbf{a})$ has c connected components for all $\mathbf{a} \in T_i\}$. As the property of having c connected components is invariant under affine transformations of the x_is ($\mathrm{AGL}_n(\mathbb{R})$), this union of T_is is also affine-invariant (under the induced action of the same $\mathrm{AGL}_n(\mathbb{R})$, not under the larger $\mathrm{AGL}_N(\mathbb{R})$). □

Remark 5.8. *The use of inequalities here is necessary. The vanishing of some test polynomials would not suffice, even when the semi-algebraic variety is defined only by equalities. This can be seen even in the simple case of the number of connected components defined by a quadratic: over $\mathbb{R}$ the number of connected components of the algebraic set $\{x \in \mathbb{R} | ax^2 + bx + c = 0\}$ is zero if and only if $b^2 - 4ac < 0$ and is at most one if and only if $b^2 - 4ac \leq 0$. The set $\{(a,b,c) | b^2 - 4ac < 0\}$ is not defined by the vanishing of some polynomials, for it has dimension 3, but the only three-dimensional subset of $\mathbb{R}^3$ defined by the vanishing of polynomials is $\mathbb{R}^3$ itself. Hence, inequalities are necessary.*

Remark 5.9. *Note that the above lower bound implies the same lower bound for decision trees for element distinctness over $\mathbb{C}$. However, over $\mathbb{C}$ the connected components argument does not work directly, because semi-algebraic varieties over $\mathbb{C}$ tend to have fewer connected components than over $\mathbb{R}$. In particular, the semi-algebraic variety corresponding to element distinctness over $\mathbb{C}$ has just a single connected component. Hence although the lower bound holds over $\mathbb{C}$, we would still only get a separating $\mathrm{AGL}_n(\mathbb{R})$-module.*

Algebraic decision trees for k-equals

Hard function: k-equals (are at least k of the inputs equal?)
Complexity class: Real semi-algebraic decision trees
Lower bound: Depth $\Omega(n \log(n/k))$ (Yao 1997)
Invariance: $\mathbb{R}$-affine ($\mathrm{AGL}_n(\mathbb{R})$), and more generally any topological homeomorphism of $\mathbb{R}^n$
Invariant property defined by polynomial inequalities: The key property used here is a lower bound on *any* Betti number, rather than just the number of connected components (=the 0-th Betti number). As the Betti numbers are invariant under homeomorphism, essentially the same argument as above using Hardt's Triviality Theorem works for this result. □

"$\mathsf{P} \neq \mathsf{NC}$" in the PRAM model without bit operations

Hard function: Max-flow with weights of bit-length $\leq O(v)$ (v=number of vertices); min-cost flow with weights of bit-length $\leq O(v)$; combinatorial linear programming[16]
Complexity class: The Boolean PRAM model without bit operations[17]

[16] Combinatorial linear programming is the following problem: given vectors b, c and a matrix A, maximize $c \cdot x$ subject to $Ax \leq b$, where the entries of A have bit-length logarithmic in the dimension; it is this latter condition that makes it "combinatorial."

[17] Briefly, the PRAM model without bit operations is as follows. It consists of non-uniform PRAMs over $\mathbb{Z}$—as in the real RAM or Blum–Shub–Smale models—but where the non-uniformity is both in the number of input parameters *and their total bit-length*, thus making it essentially a Boolean model. That is, there is a separate $\mathrm{PRAM}_{\mathbb{Z}}$ program for each pair (n, N), which gets applied to n-variate instances of bit-length at most N. Equivalently, the model

Lower bound: Parallel time $\sqrt{n}/c$ on $2^{\sqrt{n}/c}$ processors, for large enough constant c (Mulmuley 1999)
Invariance: $\mathbb{R}$-affine ($\mathrm{AGL}_3(\mathbb{R})$)
Invariant property defined by polynomial inequalities: In order to describe the key property here, we must first describe what the input objects are. An input object here is not a polynomial, but is rather the set of all integer points in $\mathbb{R}^3$ of bit-length bounded by a certain parameter b (which, in the application, will be linear in the number of inputs n), together with a label "yes" or "no" on each such integer point.

Given an arrangement of surfaces, we say that a *face* of the arrangement is a connected component of the complement. The key property Π is then as follows. A labeling of integer points as above has property $\Pi_{s,d,b}$ if there is an arrangement of s surfaces in $\mathbb{R}^3$, each of degree at most d, such that each integer point of bit-length at most b lies in a cell of the arrangement, and such that no cell contains two integer points with opposite labels.

The way Mulmuley gets from inputs in $\mathbb{Q}^n$ to inputs in $\mathbb{Z}^3 \subseteq \mathbb{R}^3$ is via parametric complexity (e.g., Carstensen 1983a,b; Mulmuley 1999; Murty 1980; Zadeh 1973, not to be confused with parametrized complexity in the sense of Downey & Fellows 1999). That is, one considers an instance of max-flow with n edges, where each capacity is a function of a single rational parameter λ, which in turn we view as a pair of integer parameters (its numerator and denominator). An instance of max flow can then be specified by giving this numerator-denominator pair and an integer threshold, hence $\mathbb{Z}^3$.

Mulmuley (1999, Theorem 5.8) shows that a PRAM without bit operations running in time $t = \sqrt{n}/c$ on 2^t processors partitions $\mathbb{R}^3$ into at most 2^{20t^2} surfaces of degree at most 2^t. To get the lower bound, Mulmuley (1999, Theorem 5.7) then shows that this is not possible for a parametrized instance of max-flow with optimal parametric complexity. The proof depends only on

Footnote 17 continued
consists of the parallel version of algebraic computation trees over $\mathbb{Z}$ as in the results of Ben-Or, Yao, *et al.*, but where the computation tree may depend not only on the number of inputs but also on their total bit-length.

the number of connected components and the number of inflection points of such an arrangement of surfaces, both of which are preserved by $\mathrm{AGL}_3(\mathbb{R})$. Again, to see that these can be defined in terms of test equalities and inequalities, we apply Hardt's Triviality Theorem.

Remark 5.10. *It is worth noting that although Mulmuley's result* (Mulmuley 1999) *uses the same Milnor–Thom bound on the number of connected components that is used by Ben-Or and Yao, there is a crucial difference. Whereas Ben-Or's and Yao's results were based on purely topological properties (of semi-algebraic varieties), Mulmuley's result is based on a* geometric *property, which is* not *preserved by topological homeomorphism, nor even by diffeomorphism nor semi-algebraic homeomorphism. In particular, inflection points will not be preserved by these more general automorphisms, giving a precise technical sense in which the PRAM result is stronger than the preceding results in the algebraic computation tree model.*

5.4. The method of approximations (finite fields). Here we give two representative examples of how results that use the method of approximation for circuits over finite fields yield separating modules. Results using similar properties, such as those of Grigoriev & Razborov (2000), should similarly yield separating modules.

Razborov–Smolensky
Hard function: MOD_3
Complexity class: $\mathsf{AC}^0[2]$
Lower bound: Exponential size (Razborov 1987; Smolensky 1987)
Invariance: $\mathbb{F}_2$-affine ($\mathrm{AGL}_n(\mathbb{F}_2)$)
Separating module: Every $\mathsf{AC}^0[2]$ circuit computes a polynomial function over $\mathbb{F}_2$, so we use $\Omega_{\mathbb{F}_2}^{d,n} \stackrel{def}{=} \mathrm{Poly}_{\mathbb{F}_2}^d(x_1, \ldots, x_n)/\langle x_1^2 = x_1, \ldots, x_n^2 = x_n\rangle$ as the space of input functions (using Ω we follow Smolensky's notation). Note that here we consider two functions equal if they are equal when evaluated on all $\mathbb{F}_2$ points. In other words, we are considering *functions* on $\mathbb{F}_2$, rather than formal polynomials whose coefficients are in $\mathbb{F}_2$. Every function

over $\mathbb{F}_2$ can be represented by a unique multilinear polynomial; when we refer to MOD_3, we mean its corresponding $\mathbb{F}_2$-multilinear polynomial.

Fix a depth k and a constant λ. For our purposes, the key property used here is:

> There exists a subset $\Gamma \subseteq \mathbb{F}_2^n$ (for "good") of size at least $2^n - 2^{n-r}$ such that f agrees with a polynomial of degree $\leq (2\lambda r)^k$ on the points in Γ.

Smolensky (1987, Lemma 2) shows that this holds for any function computed by a depth k circuit with parity gates for $r = o(n^{1/2k})$, but not for MOD_3. This condition is clearly $\mathrm{GL}_n(\mathbb{F}_2)$-invariant.

For any $\Gamma \subseteq \mathbb{F}_2^n$, let I_Γ be the ideal of polynomials that vanish everywhere on Γ. When we mod out the space of functions by I_Γ, this is the same as only considering the values a function takes on Γ. Then f agrees with a polynomial of degree $\leq d = (2\lambda r)^k$ on the points in Γ if and only if all of the coefficients of monomials of degree $> d$ of $f \pmod{I_\Gamma}$ vanish. As the map $\Omega^{d,n} \to \Omega^{d,n}/I_\Gamma$ is linear, the coefficients of $f \pmod{I_\Gamma}$ are linear combinations of the coefficients of f, and we are asking that certain such linear combinations vanish. Let T_Γ be the test module consisting of these linear combinations. Finally, for an appropriate choice of r, by Fact 5.1, $\prod_\Gamma T_\Gamma$ is the desired separating module, where the product is taken over all (finitely many) subsets $\Gamma \subseteq \mathbb{F}_2^n$ of size $\geq 2^n - 2^{n-r}$. □

Depth 3 algebraic circuits over finite fields

Hard function: Determinant

Complexity class: Depth 3 algebraic circuits over the finite field $\mathbb{F}_q$

Lower bound: Exponential size (Grigoriev & Karpinski 1998)

Invariance: $\mathbb{F}_q$-linear ($\mathrm{GL}_n(\mathbb{F}_q)$)

Separating module: As above, the key property here uses an existential quantifier over some finite collection of subsets S of $\mathbb{F}_q^n$, which will turn into a big product of test modules over all possible choices for S. Beyond that, the condition here is more complicated than above.

Here, we work in the space of formal polynomials over $\mathbb{F}_q$, namely $\mathrm{Poly}^d_{\mathbb{F}_q}(x_{11}, x_{12}, \ldots, x_{nn})$. To describe the key property, we introduce some notation. Given $\sigma \in \mathrm{GL}_n(\mathbb{F}_q)$ and any function $f = f(X)$, we denote $f(\sigma X)$ by $f^\sigma = f^\sigma(X)$. For any set F of functions, write $F^\sigma = \{f^\sigma | f \in F\}$. Let $\partial^{\leq r}(f)$ denote the linear span of all the partial derivatives of f of order $\leq r$. Finally, combining these notations, we have $\partial^{\leq r}(f)^\sigma = \{g^\sigma | g \in \partial^{\leq r}(f)\}$.

The key property of a function $f \in \mathrm{Poly}^d_{\mathbb{F}_q}(x_1, \ldots, x_n)$ is then, for appropriate choices of all the parameters involved, that there exists a subset $S \subseteq \mathrm{GL}_n(\mathbb{F}_q)$ of size $\leq s$ such that

> there is a function $g(X)$ in the intersection $\bigcap_{\sigma \in S} \partial^{\leq r}(f)^\sigma$ such that $g(A) = 0$ for all $A \in \mathrm{GL}_n(\mathbb{F}_q)$.

Again, this property is readily seen to be $\mathrm{GL}_n(\mathbb{F}_q)$-invariant. Let us verify that it is defined by test polynomials. For now, fix a subset $S \subseteq \mathrm{GL}_n(\mathbb{F}_q)$. For each $\sigma \in S$, we compute a linear basis of $\partial^{\leq r}(f)^\sigma$. The coefficients of each such basis function will be linear combinations of the coefficients of f (=test variables). This follows from the usual fact about partial derivatives, and the fact that for any $\sigma \in \mathrm{GL}_n(\mathbb{F}_q)$ and any function h, the coefficients of h^σ are linear combinations of the coefficients of h. Next, we take the intersection over all $\sigma \in S$ of these subspaces. Again, a linear basis for the resulting intersection will consist of polynomials whose coefficients are linear combinations of the test variables. Let us denote this intersection Λ.

Now observe that the collection of all g such that $g(A) = 0$ for all $A \in \mathrm{GL}_n(\mathbb{F}_q)$ is an ideal I in the space of polynomials (of degree $\leq d$ for some d), and in particular is a linear subspace thereof. Then the property is satisfied exactly if $I \cap \Lambda \neq 0$. The system of linear equations defining $I \cap \Lambda$ has coefficients which are either linear combinations of the coefficients of f (coming from the equations defining the linear space Λ) or constants (coming from the equations defining I). If this system of equations had the same number of variables as equations we could require that just the $n \times n$ determinant of the system vanishes. As the system is likely to have more equations than variables, we must require that all the $n \times n$ minors of this system vanish. These $n \times n$ minors form a test module T_S, and then, as above, the separating module is $\prod_S T_S$, where the product is over all S of appropriate size.

Remark 5.11. *Aside from the more obvious uses of finiteness (not just finite characteristic) in the above proofs, in the Grigoriev–Karpinski proof, the property they use becomes vacuous over any infinite field $\mathbb{F}$: The only polynomial in n^2 variables that vanishes everywhere on $\mathrm{GL}_n(\mathbb{F})$ is the zero polynomial. For further discussion of these issues, see Section 4.5.*

5.5. Results already known to give a separating module.

Permanent versus determinant

Hard function: perm_n
Complexity class: Linear projections of $\det_m$
Lower bound: $m \geq n^2/2$ (Mignon & Ressayre 2004); also border determinantal complexity $n^2/2$ (Landsberg, Manivel & Ressayre 2013)[18]
Invariance: $\mathbb{C}$-linear ($\mathrm{GL}_{m^2}(\mathbb{C})$)
Separating module: The key property used here is the rank of the Hessian matrix of a function. Recall that the Hessian of a function $f(x_1, \ldots, x_n)$ is the $n \times n$ matrix $\mathrm{Hess}(f)$ whose (i, j) entry is the second partial derivative $\partial^2 f/\partial x_i \partial x_j$. Mignon & Ressayre (2004) show a lower bound on $\mathrm{rk}\,\mathrm{Hess}(\mathrm{perm})$ and an upper bound on $\mathrm{rk}\,\mathrm{Hess}(\det)$. Note that the entries of $\mathrm{Hess}(\det)$ are themselves functions; the upper bound on $\mathrm{rk}\,\mathrm{Hess}(\det)$ that they prove does not hold at all matrices X, but only at those matrices where $\det(X) = 0$. This is enough for them to prove the lower bound, but it took additional work to turn it into a separating module (Landsberg, Manivel & Ressayre 2013).

If the upper bound held for all X, then the minors of the Hessian matrix would span a separating module, as in the Nisan–Wigderson partial derivatives technique above. Instead, the condition they use is that $\det(X)$ divides the $r \times r$ minors of $\mathrm{Hess}(\det)$ (for $r = 2n + 1$). Landsberg *et al.* (2013) find polynomial equations that vanish exactly on the pairs of homogeneous polynomials (f, g) such

[18] The way Mignon and Ressayre phrased their result, it only applied to irreducible polynomials, and hence could not be extended to a separating module *mutatis mutandis*. Landsberg, Manivel, and Ressayre gave a more general result which allowed one to extract a separating module based on a geometric interpretation of the underlying key property.

that f divides g (among other achievements); they then construct a separating module by using these equations with $f = \det$ and g the $r \times r$ minors of $\mathrm{Hess}(\det)$.

Let D (for "divides") be the set of pairs of homogeneous polynomials $(f, g) \in \mathrm{Poly}^d \times \mathrm{Poly}^e$ such that f divides g. Landsberg, Manivel & Ressayre (2013, Section 2.3) give test polynomials defining D by considering restrictions to two-dimensional subspaces; here we give an alternative, and we believe slightly more direct, proof. Let $\mathrm{Mult}_f \colon \mathrm{Poly}^{e-d} \to \mathrm{Poly}^e$ be the linear map given by multiplying a homogeneous polynomial of degree $e-d$ by f; then f divides g if and only if g is in the image of Mult_f. Let M_f be the $(\dim \mathrm{Poly}^{e-d}) \times (\dim \mathrm{Poly}^e)$ matrix corresponding to Mult_f (say, in the basis of monomials), and let c_g denote the column vector consisting of the coefficients of g. Then, correspondingly, f divides g if and only if c_g lies in the linear span of the columns of M_f. The latter happens if and only if the exterior product of the columns of M_f and c_g vanishes, or equivalently, all of the maximal minors of the matrix $(M_f | c_g)$ consisting of M_f with an additional column c_g added on vanish. It is clear that each entry of M_f is either 0 or a coefficient of a single monomial of f, so these equations are indeed test polynomials in the coefficients of f and g.

Now, we consider the preceding equations with g replaced by each of the $r \times r$ minors of $\mathrm{Hess}(f)$, in turn. For any such $r \times r$ minor, this furnishes a set of test polynomials in the coefficients of f alone, and we consider the union of all these sets, over the choice of all $r \times r$ sub-matrices of $\mathrm{Hess}(f)$.

For an irreducible, or even square-free, polynomial f, the condition that f divides g is equivalent to g vanishing wherever f vanishes. Thus, combining the preceding equations with the original result of Mignon & Ressayre (2004, Proposition 3.8) shows that the above test module vanishes at $f = \det_m$ for $r = 2n+1$.

Next, to be precise, note that when we compare $\det_m$ with perm_n, we will in fact compare $\det_m$ with

$$\overline{\mathrm{perm}}_{m,n}(X) := x_{mm}^{m-n}\, \mathrm{perm}_n(X|_n),$$

where $X|_n$ is the upper-left $n \times n$ sub-matrix of the $m \times m$ matrix X. From the complexity viewpoint, this essentially changes nothing, but allows us to work entirely within the space of degree m homogeneous polynomials in m^2 variables.

Now, despite the fact that $\overline{\mathrm{perm}}_{m,n}$ is not square-free (except for the relatively easy case of $m = n+1$), we show that the preceding test polynomials nonetheless capture the argument of Mignon–Ressayre (even if they cannot be used to define the collection of varieties with degenerate duals, as considered by Landsberg–Manivel–Ressayre, as there is no known workable definition of the dual variety of a variety defined by a reducible and/or non-square-free polynomial). That is, we want to show that $\overline{\mathrm{perm}}_{m,n} = x_{mm}^{m-n}\,\mathrm{perm}_n(X|_n)$ does not divide some $r \times r$ minor of $\mathrm{Hess}(\overline{\mathrm{perm}}_{m,n})$ for some $r \geq 2n+1$. To show this, we describe part of this Hessian explicitly.

Recall that the variables here are indexed by pairs (i, j), so a second partial derivative—an entry of the Hessian—is indexed by a pair of pairs, such as $((i, j), (i', j'))$. For such pairs with $i, j, i', j' \leq n$, the $((i, j), (i', j'))$ entry of $\mathrm{Hess}(\overline{\mathrm{perm}}_{m,n})$ is just x_{mm}^{m-n} times the same entry of $\mathrm{Hess}(\mathrm{perm}_n(X|_n))$. In particular, an $r \times r$ minor of $\mathrm{Hess}(\overline{\mathrm{perm}}_{m,n})$ consisting entirely of such entries is just $x_{mm}^{r(m-n)}$ times the same minor $M(X|_n)$ of $\mathrm{Hess}(\mathrm{perm}_n(X|_n))$. Since both $\mathrm{perm}_n(X|_n)$ and $M(X|_n)$ do not depend on x_{mm}, and $r > 0$, note that $x_{mm}^{m-n}\,\mathrm{perm}_n(X|_n)$ divides $x_{mm}^{r(m-n)} M(X|_n)$ if and only if $\mathrm{perm}_n(X|_n)$ divides $M(X|_n)$. Since perm_n is indeed irreducible, we are in the same situation we were in above, and we can show that $\mathrm{perm}_n(X|_n)$ does not divide $M(X|_n)$ by exhibiting an $n \times n$ matrix A such that $\mathrm{perm}_n(A) = 0$ but $M(A) \neq 0$. Such a matrix A is given exactly by Mignon & Ressayre (2004, Proposition 3.3), for $r = n^2$. Hence, $\overline{\mathrm{perm}}_{m,n}$ does not divide some $r \times r$ minor of $\mathrm{Hess}(\overline{\mathrm{perm}}_{m,n})$ for $r = n^2 > 2n + 1$, and the (polynomial) test module above does not vanish at $\overline{\mathrm{perm}}_{m,n}$ for $m < n^2/2$. □

Matrix multiplication

Hard function: $n \times n$ matrix multiplication

Complexity class: Bilinear circuits in characteristic zero

Lower bound: Border rank $\geq 2n^2 + o(n^2)$ (Landsberg & Ottaviani 2011) (independently, $\geq \frac{3}{2}n^2 - 2$ Bürgisser & Ikenmeyer 2013)

Invariance: $\mathbb{F}$-linear $(\mathrm{GL}_{n^2}(\mathbb{F}) \times \mathrm{GL}_{n^2}(\mathbb{F}) \times \mathrm{GL}_{n^2}(\mathbb{F}))$, characteristic zero

Separating module: Landsberg and Ottaviani construct separating modules yielding this lower bound. The bound of Bürgisser and Ikenmeyer, although weaker, constructs "occurrence obstructions," which are more stringent than separating modules, and more in line with the "full" GCT Program; see Section 4.1 for the definition.

6. Relations between lower bounds yield relations between separating modules

Baur–Strassen: computing partial derivatives (1983)
Assumption: Computing $(\partial f/\partial x_1, \ldots, \partial f/\partial x_n)$ requires algebraic circuits of size s
Consequence: Computing f requires algebraic circuits of size $s/3$
Invariance: $\mathbb{F}$-linear ($\mathrm{GL}_n(\mathbb{F})$), any infinite field
Separating module implication: Let φ be the map from $\mathrm{Poly}^d(\vec{x})$ to the Chow variety or Hilbert scheme (see The Degree Bound above), defined as follows. $\varphi(f)$ is the variety (ideal) defined by $\langle \partial f/\partial x_1, \ldots, \partial f/\partial x_n \rangle$. Recall that $A \in \mathrm{GL}_n(\mathbb{F})$ acts on the Hilbert scheme by taking the ideal $\langle g_1(\mathbf{x}), \ldots, g_k(\mathbf{x})\rangle$ to the ideal $\langle g_1(A\mathbf{x}), \ldots, g_k(A\mathbf{x})\rangle$; denote the latter by $A \cdot \langle g_1(\mathbf{x}), \ldots, g_k(\mathbf{x})\rangle$. Similarly, $A \in \mathrm{GL}_n(\mathbb{F})$ acts on $\mathrm{Poly}^d(\mathbf{x})$ by sending $f(\mathbf{x})$ to $f(A\mathbf{x})$. Then φ is $\mathrm{GL}_n(\mathbb{F})$-equivariant, in that

$$\begin{aligned}\varphi(f(A\mathbf{x})) &= \left\langle \sum_j a_{1j}\left(\frac{\partial f}{\partial x_j}\right)(A\mathbf{x}), \ldots, \sum_j a_{nj}\left(\frac{\partial f}{\partial x_j}\right)(A\mathbf{x})\right\rangle \\ &= \left\langle \left(\frac{\partial f}{\partial x_1}\right)(A\mathbf{x}), \ldots, \left(\frac{\partial f}{\partial x_n}\right)(A\mathbf{x})\right\rangle \\ &= A \cdot \varphi(f(\mathbf{x})).\end{aligned}$$

If T is a test module that vanishes on

$$\{\varphi(g) | \varphi(g) \text{ has algebraic circuits of size } \leq s\},$$

but not on $\varphi(f)$, then $\varphi_*(T) \overset{def}{=} \{t \circ \varphi | t \in T\}$ is a vector space of test polynomials which vanishes at all $g \in \mathrm{Poly}^d(\mathbf{x})$ that have circuits of size $\leq s/3$, but not at f. The $\mathrm{GL}_n(\mathbb{F})$-equivariance of φ implies that $\varphi_*(T)$ is in fact a test $\mathrm{GL}_n(\mathbb{F})$-module.

Tensor rank to formula size (Raz 2010b)
Assumption: $t_n \in (\mathbb{F}^n)^{\otimes r(n)}$ has tensor rank $\geq n^{r(n)(1-o(1))}$ for some $\omega(1) \leq r(n) \leq O\left(\frac{\log n}{\log \log n}\right)$
Consequence: The polynomial f_n which is the symmetrization of t_n requires super-polynomial size algebraic formulas. Also, by the completeness of the permanent, perm_n requires super-polynomial size algebraic formulas (attributed to Yehudayoff in Raz 2010b, Footnote 2)
Invariance: $\mathbb{F}$-linear ($\text{GL}_n(\mathbb{F})$), $\mathbb{F}$ arbitrary
Separating module implication: Raz uses the standard symmetrization map from tensors $(\mathbb{F}^n)^{\otimes r}$ (we think of these as degree r homogeneous non-commutative polynomials) to $\text{Poly}^r(x_1, \dots, x_n)$. In particular, to show an algebraic formula size lower bound on some $f_n \in \text{Poly}^r(\mathbf{x})$, it suffices to show a tensor rank lower bound on *any* non-commutative version t_n of f_n (that is, f_n is the result of symmetrizing t_n). In particular, we are free to use the standard embedding (NB: in the opposite direction compared to the above) $\varphi\colon \text{Poly}^r(\mathbf{x}) \hookrightarrow (\mathbb{F}^n)^{\otimes r}$, which takes the monomial $x_{i_1} \dots x_{i_r}$ to the tensor $\frac{1}{r!}\sum_{\pi \in S_r} x_{i_{\pi(1)}} \otimes \cdots \otimes x_{i_{\pi(r)}}$. Raz's results imply that the image, under φ, of the set of polynomials that have small formulas is contained in the set of tensors of low tensor rank. It is a standard fact from multi-linear algebra that the embedding φ is $\text{GL}_n(\mathbb{F})$-equivariant (see the Baur–Strassen implication above). Hence, if a test module T is used to show a lower bound on the tensor rank (and hence, border rank, see Appendix B.1) of some $\varphi(f)$, then $\{t \circ \varphi | t \in T\}$ is a test module which implies the stated lower bound on the algebraic formula size of f.

Chasm at Depth 4 (Agrawal & Vinay 2008; Koiran 2012; Tavenas 2013)
Assumption: f requires depth 4 algebraic circuits of size $2^{\omega(\sqrt{n}\log n)}$
Consequence: f requires algebraic circuits of super-polynomial size
Invariance: $\mathbb{F}$-affine ($\text{AGL}_n(\mathbb{F})$), $\mathbb{F}$ arbitrary
Separating module implication: They show that the set of functions computable by algebraic circuits of polynomial size is contained in the set of functions computable by depth 4 circuits of

size $2^{O(\sqrt{n}\log n)}$. Hence, if a separating module vanishes on the latter set, it also vanishes on the former.

Chasm at Depth 3 (Gupta, Kamath, Kayal & Saptharishi 2013; Tavenas 2013)
Assumption: f requires depth 3 algebraic circuits of size $2^{\omega(\sqrt{n}\log n)}$
Consequence: f requires algebraic circuits of super-polynomial size
Invariance: $\mathbb{F}$-affine ($\mathrm{AGL}_n(\mathbb{F})$), characteristic zero or characteristic $> \deg f$
Separating module implication: Same as above, but not over arbitrary fields. See Section 4.5 for a discussion of this issue.

Matrix rigidity to linear circuits (Valiant 1977)
Assumption: The $n \times n$ matrix A_n has rigidity $R_{A_n}(n/2) \geq \Omega(n^{1+\varepsilon})$
Consequence: The linear function $\mathbf{x} \mapsto A_n\mathbf{x}$ does not have linear circuits of simultaneous size $O(n)$ and depth $O(\log n)$
Invariance: permutation ($S_n \times S_n$)
Separating module implication: Here the ambient (input) space is the space $M_n(\mathbb{F})$ of $n \times n$ matrices. Valiant (1977, Corollary 6.3) showed the set of matrices A_n whose associated linear functions $x \mapsto A_n x$ can be computed by linear circuits of size $O(n)$ and depth $O(\log n)$ (simultaneously) is contained in the set of matrices of low rigidity. Hence, any test module which vanishes on the set of matrices with low rigidity but not on some matrix A will also vanish on the set of matrices that can be computed in size $O(n)$ and depth $O(\log n)$ by linear circuits.

As the concept of rigidity involves the *number of entries* of a matrix that must be changed to drop its rank, this concept is only permutation-invariant—we may multiply A_n on the left and right by permutation matrices without affecting its rank or rigidity. We note that, despite the fact that the non-rigid matrices do not form an algebraic set, the most successful results on matrix rigidity to date use the algebro-geometric approach (essentially, test polynomials) (Gesmundo, Hauenstein, Ikenmeyer & Landsberg 2013; Kumar, Lokam, Patankar & Sarma 2009; Landsberg, Taylor & Vishnoi 2003).

Acknowledgements

The author would like to thank Scott Aaronson, Eric Allender, Saugata Basu, Tom Church, Klim Efremenko, Kaveh Ghasemloo, J. M. Landsberg, Ketan Mulmuley, Toni Pitassi, Peter Scheiblechner, Chris Umans, Alasdair Urquhart, Ryan Williams, and Yiwei She for useful discussions. In particular, Williams suggested the terminology "input polynomial vs. test polynomial," and Shipman and Church helped come up with the name "separating module." The author also had several useful conversations with Shipman regarding the Hilbert scheme, used to show the degree bound fits into this framework. Basu pointed the author to Hardt's Triviality Theorem, used to show the connected components sorting lower bound fits into this framework. Discussions with Scheiblechner were useful for Section 4.3. The author had useful discussions with Ghasemloo and Pitassi regarding whether and how the Razborov–Smolensky result fits into this framework. The author would also like to thank Amir Yehudayoff for his wonderfully clear talk on multilinear lower bounds at Dagstuhl. Some of these conversations were facilitated by the Dagstuhl seminar on Algebraic and Combinatorial Methods in Computational Complexity in October 2012 (thanks to the organizers Manindra Agrawal, Thomas Thierauf, and Chris Umans for the invitation), and by the Exploring the Limits of Computation Tokyo Complexity Workshop in March 2013 (thanks to the organizers Osamu Watanabe and Jun Tarui for the invitation and support). This paper would not have been possible without many useful conversations with Ketan Mulmuley over several years. Toni Pitassi provided very helpful comments on a draft. The author thanks several anonymous reviewers for useful comments. The author would also like to thank Allan Borodin for his support, both material and otherwise, part of which came by way of Borodin's NSERC Grant #482671. An extended abstract of this paper has appeared as Grochow (2014).

A. Proof of the correspondence between invariant properties and test modules

Fact A.1 (Generalized restatement of Fact 2.8). *Let G be any group (not necessarily finite) acting linearly on an input vector space. There is a many-to-one correspondence between test G-modules and G-invariant properties defined by the vanishing of test polynomials.*

Proof. Let V denote the input space (input polynomials, matrices, etc.), and suppose that T is a test G-module with basis $t_1, \dots, t_k$. Let Π_T denote the corresponding property, namely $\Pi_T = \{v \in V | t(v) = 0 \forall t \in T\}$. Π_T is defined by test polynomials (namely, those in T). To see that Π_T is G-invariant, suppose that $v \in \Pi_T$ and $g \in G$, and consider the point gv. By the defining property of test G-module, if $t(\mathbf{x}) \in T$, then $t(g\mathbf{x}) \in T$ for all $g \in G$. Let $t'(\mathbf{x}) = t(g\mathbf{x})$. As $t' \in T$ and $v \in \Pi_T$, we have $t'(v) = 0$ by the definition of Π_T. But then $0 = t'(v) = t(gv)$, as desired. Hence, Π_T is a G-invariant property defined by test polynomials.

Conversely, suppose that $\Pi \subseteq V$ is a G-invariant property defined by test polynomials. By Hilbert's Basis Theorem, Π is defined by the vanishing of only finitely many test polynomials, say $t_1, \dots, t_k$. If G is finite, then it is clear that the collection of polynomials $\{t_i(g(\mathbf{x}))|1 \leq i \leq k, g \in G\}$ is finite, and hence its linear span, which we denote GT, is finite-dimensional. More generally, for arbitrary groups G, since we have assumed that the action of G on V is linear, the map $t(\mathbf{x}) \mapsto t(g\mathbf{x})$ preserves the degree of t for any $g \in G$. Thus, GT is finite-dimensional, as it is a subset of the polynomials in $\mathbf{x}$ of degree at most $\max_i \deg t_i$.

We will show that for arbitrary Π defined by the test polynomials in T (not necessarily G-invariant), Π_{GT} is the unique maximum G-invariant subset of Π. Hence, if Π itself is G-invariant, then $\Pi = \Pi_{GT}$. Suppose Π' is a G-invariant subset of Π. In particular, every test polynomial $t \in T$ vanishes on every $v \in \Pi'$. We must show that for arbitrary g, $t(g\mathbf{x})$ also vanishes on every $v \in \Pi'$. As Π' is G-invariant, $v \in \Pi'$ implies that $gv \in \Pi'$ for every $g \in G$. Hence, $t(gv) = 0$ for every $v \in \Pi'$. Thus, $\Pi' \subseteq \Pi_{GT}$. As this holds for arbitrary G-invariant subsets Π' of Π, Π_{GT} is the unique maximum G-invariant subset of Π, and thus is equal to Π if Π itself is G-invariant. □

It is clear that the map sending a test G-module T to the property Π_T is well defined, and hence is at worst many-to-one. Over an algebraically closed field, Hilbert's Nullstellensatz implies that two test G-modules T_1 and T_2 define the same property Π if and only if they generate the same radical ideal. Hence, we cannot expect this map to be one-to-one.

B. More on the necessity and utility of separating modules

B.1. Test polynomials & border/approximative complexity. Over any field, if $\mathcal{C}_n$ is defined by test polynomials, say $\mathcal{C}_n = \{f | t_1(f) = t_2(f) = \cdots = t_k(f) = 0\} \subseteq \mathrm{Poly}^{d(n)}(x_1, \ldots, x_n)$, then $f_{hard,n} \notin \mathcal{C}_n$ if and only if there is some $1 \leq i \leq k$ such that $t_i(f_{hard}) \neq 0$. For such classes, the use of test polynomials is necessary and sufficient to prove a lower bound. However, most complexity classes are not defined by test polynomials in this manner. Hence when we prove a lower bound against $\mathcal{C}_n$ using test polynomials, we in fact prove a lower bound against the slightly larger class which we denote $\overline{\mathcal{C}_n}$ and refer to as "border-$\mathcal{C}_n$" or "approximative $\mathcal{C}_n$" in line with normal usage in other contexts (the overline is for Zariski closure; see Definition B.4). Standard results in algebraic geometry (e.g., Mumford 1976, Theorem 2.33, Bürgisser, Clausen & Shokrollahi 1997, Section 20.6) imply that $\overline{\mathcal{C}_n}$ consists of all functions f which can be written as a limit of functions in $\mathcal{C}_n$ (see Footnote 9 on page 409).

Our thesis here, that we hope to convince the reader of, is that border complexity in general—not only in the context of matrix multiplication—is a natural and useful measure of complexity from the perspective of lower bounds. The validity of this idea was perhaps first recognized in 1974, by V. Strassen, who wrote (in the following quotation $\overline{L}$ denotes border circuit size and L denotes circuit size):

> "$\overline{L}$ is more manageable than L: e.g., the knowledge of L is equivalent to the knowledge of the formulas [defining the] constructible sets [$\mathcal{C}$ (in our notation)], whereas $\overline{L}$ is known if the much simpler closed sets [$\overline{\mathcal{C}}$] or the corresponding polynomial ideals are known." — Strassen (1974, p. 132)

In the remaining sections of this appendix, we argue that proving lower bounds against border-$\mathcal{C}$ is likely to be the easiest way to prove lower bounds against $\mathcal{C}$, despite being a formally stronger statement. We remark that decades of experience in algebraic geometry already suggests this to be the case. Several such arguments are leveraged in the original papers of Mulmuley & Sohoni (2001, 2008) to give such evidence specifically in the context of GCT, but those arguments are beyond the scope of this paper. The arguments we give here are meant to add to this evidence—strengthening the wisdom from algebraic geometers—and to give an intuitive complexity-theoretic viewpoint on why studying border-$\mathcal{C}$ should be useful.

We begin with some discussion and two examples where we have some idea of the difference between complexity and border complexity; in the case of matrix multiplication, there turns out to be no difference of asymptotic consequence.

Example B.1 (Matrix multiplication). In the context of matrix multiplication, the typical complexity measure is *tensor rank*, which is essentially the number of non-scalar multiplications needed to multiply two matrices. Tensor rank is known to agree with the total number of algebraic operations up to a constant factor. The corresponding border complexity measure is called *border rank* or "approximative complexity," first introduced in this setting by Bini, Capovani, Romani & Lotti (1979). In general, border rank can be smaller than tensor rank. However, Bini (1980) showed that the exponent of matrix multiplication calculated with tensor rank—the smallest ω such that $n \times n$ matrix multiplication has tensor rank $O(n^{\omega})$—is the same as the exponent calculated with border rank. Thus, although border rank and tensor rank are not equal, they give the same asymptotic answer for matrix multiplication.

Furthermore, the use of border rank has greatly increased our understanding of both upper and lower bounds for matrix multiplication. One of the main tools for finding efficient algorithms for matrix multiplication (Coppersmith & Winograd 1990; Stothers 2010; Vassilevska Williams 2012) is Schönhage's asymptotic sum inequality (Schönhage 1981), which shows that an upper bound on border rank implies an upper bound on tensor rank. Conversely, most lower bounds on matrix multiplication seem to have a border rank lower bound at their heart. For example, Landsberg (2008, Section 6) showed that the tensor rank lower bound of Bläser (1999)—the then best known bound—implicitly uses the same key lemma that Strassen (1983) used to give a border rank lower bound. The currently best known lower bound on tensor rank (Landsberg 2014b; Massarenti & Raviolo 2012) also uses techniques from the best known lower bound on border rank (Landsberg & Ottaviani 2011). $\Diamond$

Example B.2 (Permanent versus determinant). In the context of permanent versus determinant, the typical complexity measure is *determinantal complexity*: the size of the smallest matrix $M(\mathbf{x})$ with linear combinations of the variables $\mathbf{x}$ for entries such that $\det(M(\mathbf{x})) = \mathrm{perm}(\mathbf{x})$. Mulmuley & Sohoni (2001) use the analogous notion of border determinantal complexity, which they refer to as "infinitesimal approximative" complexity. Independently, Bürgisser *et al.* (2011b, Proposition 9.4.3) and the author (Grochow 2012, Proposition 3.5.4) show that under certain fairly general circumstances the border determinantal complexity only differs from the determinantal complexity by a polynomial, and ask a question whose affirmative answer would imply this is always the case. Thus, border complexity in this context may not be as far from standard complexity as it at first seems.

On the other hand, Mulmuley & Sohoni (2001, Section 4.2) give an example of a function which has border determinantal complexity poly(n) but which may have super-polynomial determinantal complexity (if it does, this would imply a negative answer to

Bürgisser *et al.* (2011b, Open Question 9.4.2). Such functions exhibit a difference in the difficulties of resolving the complexity of matrix multiplication and resolving the permanent versus determinant problem. Nonetheless, they conjecture (Mulmuley & Sohoni 2001, Conjecture 4.3) that no VNP-hard function has polynomial border determinantal complexity. Given the VQP-completeness of the determinant, it is also natural to ask whether there is any difference in terms of quasi-polynomial determinantal versus border-determinantal complexity:

Open Question B.3. *Does polynomial or quasi-polynomial border determinantal complexity imply quasi-polynomial determinantal complexity? Equivalently, is* $\overline{\mathsf{VP}}_{ws} \subseteq \mathsf{VQP}$ *or more strongly* $\overline{\mathsf{VQP}} = \mathsf{VQP}$*?*

Recall that VQP consists of polynomials of polynomial degree that can be computed by quasi-polynomial size algebraic circuits. The determinant is complete for VQP under quasi-polynomial projections and is complete for VP_{ws} under p-projections (this can be taken as a definition of VP_{ws}, see, e.g., Malod & Portier 2008). ◊

Either way, as all of our current techniques give bounds on border complexity, Open Question B.3 is an archetype of a fundamental question of the difference between the way complexity classes are usually defined and the methods we use for proving lower bounds against them.

B.2. Non-uniform complexity classes are constructible by test polynomials. Although non-uniform complexity classes are typically not defined by test polynomials, in this section we show that all naturally occurring complexity classes (and more) are nonetheless "constructible" by test polynomials (definition below). This idea is already implicit in the works of Strassen (1974), Heintz & Sieveking (1980), Heintz & Schnorr (1982), Raz (2010a), Mulmuley & Sohoni (2001, 2008), and possibly others, but to the best of our knowledge has not been developed before so systematically as here. In the following sections, we give several arguments that test polynomials—and hence, via Lemma 3.1 and Fact 2.8, separating modules—are nonetheless useful for understanding such constructible (invariant) complexity classes.

Definition B.4 (Zariski, i.e., algebro-geometric, topology). *A set defined by the vanishing of test polynomials is called* (Zariski) closed. *A set is* constructible *if it can be constructed from closed sets by taking complements, finite unions, and finite intersections.*

The *closure* of a set S is the smallest closed set containing S and is denoted $\overline{S}$. If S is a Zariski-constructible set over $\mathbb{C}$, then its Zariski closure coincides with its closure in the usual Euclidean complex topology (see, e.g., Mumford 1976, Theorem 2.33). Note that the closure $\overline{S}$ is the set of all points which cannot be separated from S by test polynomials.

The main insight of this section is a corollary to Chevalley's Constructibility Theorem. To state this theorem, we need one more concept. A map $\varphi \colon A \to B$ between closed sets is called *algebraic* if its graph $\{(a, \varphi(a)) | a \in A\}$ is a closed subset of $A \times B$. Equivalently, if B is a Zariski-closed subset of some $\mathbb{F}^m$, then φ is algebraic if and only if for each $1 \leq i \leq m$, the coordinate $x_i(\varphi(a))$ can be expressed as a polynomial in the coordinates of $a \in A$.

Chevalley's Theorem is most concisely stated for Noetherian rings, but we will not need their definition here. For our purposes, it suffices that this includes $\mathbb{Z}$, $\mathbb{Z}/n\mathbb{Z}$, rings of algebraic integers, all fields, polynomial rings, and quotients of polynomial rings.

Theorem B.5 (Chevalley's Theorem).[19] *Over any Noetherian ring the image of any algebraic map is constructible.*

All non-uniform complexity classes we are aware of—be they algebraic or otherwise—belong to one of the classes described in the following corollary:

Corollary B.6. *Let $\mathcal{C}$ be a non-uniform complexity class; then $\mathcal{C}_n$ is (Zariski)constructible if any of the following hold:*

[19] The original version of this theorem over algebraically closed fields is from Chevalley & Cartan (1955–1956). The general version, which is in fact more general than stated here, can be found as Grothendieck (1964, Theorem 1.8.4). See Eisenbud (1995, Corollary 14.7) for a purely ring-theoretic treatment of what is essentially the general case, or Matsumura (1980, Chapter 1, Section 6).

(i) $|\mathcal{C}_n|$ *is finite; or*

(ii) $\mathcal{C}$ *is closed under simple (resp. linear, resp. affine) projections, and contains a problem that is complete under simple (resp. linear, resp. affine) projections; or*

(iii) $\mathcal{C}_n$ *is defined by a class of circuits that are restricted to have one of finitely many (a number which may grow with* n*) shapes. Here by the "shape" of a circuit, we mean the underlying directed acyclic graph together with operators labeling the internal nodes; or*

(iv) More generally, $\mathcal{C}_n$ *is first-order definable in the language of rings over a Noetherian ring, or in the language of ordered rings over an ordered Noetherian ring.*

A "simple projection" here means any map that sends each variable x_i to a constant α or to a constant multiple of a variable αy_j. A linear projection sends each x_i to a linear combination of variables $\sum_j \alpha_{ij} y_j$, and an affine projection additionally allows an additive constant: $x_i \mapsto \alpha_i + \sum_j \alpha_{ij} y_j$.

Condition (iii) includes circuit classes defined in terms of fan-in, size, depth, or connectivity properties like skew or weakly skew.

Proof. (i) Any finite set is defined by the vanishing of test polynomials; i.e., it is closed, hence constructible.

(ii) The set of simple (resp. linear, resp. affine) projections is closed, as we show below; denote this set by R, for "reductions." If f_n is a complete function, and F is the space of input functions (objects, etc.), then define a map $\varphi\colon R \to F$ by $\varphi(r) = r(f_n)$. From the definition of projection, it is easily seen that φ is algebraic. Then, $\mathcal{C}_n$ is the image of φ, and hence is constructible by Chevalley's Theorem.

The set of linear (resp. affine) projections from functions on n variables to functions on m variables is just the set of $m \times n$ (resp. $(m+1) \times n$) matrices, so is closed. The set of simple projections is the subset of affine projections defined by the property that each column of the $(m+1) \times n$ matrix has at most one nonzero entry. The latter condition is equivalent to the condition that the product of any two entries from a given column vanishes, and hence, the set of simple projections is closed.

(iii) For each circuit shape G, the set of circuits of that shape is $\mathbb{F}^N$ where N is the number of edges whose endpoints are linear combination gates. Let Ckt_G denote this space, and let $\varphi_G \colon \mathrm{Ckt}_G \to \mathrm{Poly}^d(x_1, \ldots, x_n)$ be the map which takes each circuit of shape G to the function it computes. It is easily seen that φ_G is algebraic, so its image is constructible by Chevalley's Theorem. Then, $\mathcal{C}_n$ is the union over finitely many shapes G of $\mathrm{Im}(\varphi_G)$. As a union of constructible sets is constructible, so is $\mathcal{C}_n$.

(iv) A first-order definable set is defined by some first-order formula. For quantifier-free formulas, this is exactly a set defined by a logical combination of equalities and inequalities, namely a constructible set. The only tricky part is then to handle quantifiers. By replacing a universal quantifier $\forall x$ by $\neg\exists x\neg$ and noting that the complement of a constructible set is constructible, we need only handle existential quantifiers. If $\varphi(\mathbf{x})$ is a first-order formula without quantifiers, let $\mathcal{C}'$ denote the set of those $\mathbf{x}$ that satisfy $\varphi(\mathbf{x})$. Then, the set defined by $\exists x_0 \varphi(\mathbf{x})$ is equal to the image of $\mathcal{C}'$ under the projection that sends $(x_0, x_1, \ldots, x_n)$ to $(x_1, \ldots, x_n)$. By Chevalley's Theorem, the image of this projection is constructible. □

Note that if a circuit class is defined as the image of some map—as nearly all of them are, as in conditions (ii) and (iii)—finding its representation as a union of differences of closed sets may be difficult, even uncomputable. However, over finite fields this is a finite problem, hence computable, and over algebraically closed fields or real closed fields quantifier elimination algorithms such as Tarski's (Tarski 1948) make this process effective. However, even in these cases, there may not be a description that is uniform in n, and even if there is, finding such a description may be difficult.

Remark B.7. *The (Zariski)closure of classes satisfying* Corollary B.6(ii) *for linear or affine projections are* orbit closures *for* GL_n*, respectively,* AGL_n*. Much of the current research in GCT studies the orbit closures associated with the permanent, determi-*

nant, and matrix multiplication. Considering their structure as orbit closures rather than just G-invariant sets facilitates, their study greatly much as the existence of complete problems facilitates the study of a complexity class. In this paper, we show that by extending our viewpoint to all G-invariant complexity classes and not just orbit closures, GCT becomes much more general and far-reaching.

B.3. The utility of separating modules: complexity aspects. Lemma 3.1 and Observation 3.2 show that invariant properties can be used to prove lower bounds without loss of generality. In the previous section, we showed that for all naturally occurring non-uniform complexity classes $\mathcal{C}$, $\mathcal{C}_n$ is constructible, and furthermore is typically the image of some simple algebraic map from some $\mathbb{F}^N$. We now give a heuristic argument that the easiest way to prove a lower bound against such sets is by using a test polynomial, and hence, for nearly all classes (the invariant ones), a separating module (Lemma 3.1 and Fact 2.8). Even when the use of separating modules is not formally necessary, it thus helps illuminate any (constructible) non-uniform complexity class.

Here we argue based essentially on dimension and the complexity of the description of constructible sets in terms of unions, intersections, and complements of closed sets. In the next section, we give a more nuanced argument based on the geometric and computational properties of the boundaries of these constructible sets.

If $\mathcal{C}_n$ is closed, then we said above that the use of test polynomials is necessary and sufficient to prove $f_{hard,n} \notin \mathcal{C}_n$. For the sake of discussion, suppose that $\mathcal{C}_n$ is not closed, but is the next simplest kind of constructible set: $\mathcal{C}_n$ is locally closed; i.e., is equal to the difference $\mathcal{A}_n \backslash \mathcal{B}_n$ for some closed sets $\mathcal{A}_n, \mathcal{B}_n$. Without loss of generality, we may assume that $\mathcal{A}_n = \overline{\mathcal{C}_n}$ is the Zariski closure of $\mathcal{C}_n$, and that $\mathcal{B}_n \subseteq \mathcal{A}_n$.

This immediately suggests two approaches to show $f_{hard,n} \notin \mathcal{C}_n$:

(1) Show that $f_{hard,n} \notin \mathcal{A}_n = \overline{\mathcal{C}_n}$; or

(2) Show that $f_{hard,n} \in \mathcal{B}_n$.

Note that in (2), we are not assuming that approach (1) has failed; i.e., that $f_{hard,n}$ is in fact in $\overline{\mathcal{C}_n}$; we are rather considering a completely alternative approach, independent of the success or failure of approach (1). See Footnote 22 for why this distinction is relevant.

As $\mathcal{B}_n = \overline{\mathcal{C}_n} \backslash \mathcal{C}_n$ might be complicated, a third approach is:

(3) Find a closed set $\mathcal{D}_n$ containing $f_{hard,n}$ such that $\mathcal{D}_n$ is disjoint from $\mathcal{C}_n$.

Remark B.8 (Elusive functions). *Although the elusive functions of* Raz (2010a) *are essentially a special case of approach (3), in fact all the lower bounds that have currently been shown using elusive functions* (Lê 2010, 2013; Raz 2010a) *find such a closed set $\mathcal{D}_n$ that is disjoint not only from $\mathcal{C}_n$ but from its closure (see* Section 5.2*). Thus, the current lower bounds that are proved using elusive functions, despite having the philosophy of approach (3), are in fact using test polynomials as in approach (1).*

Remark B.9 (Algebraic pseudorandom generators). *As with elusive functions, the algebraic pseudorandom generators introduced by* Agrawal (2005)[20] *are again essentially a special case of approach (3), but known applications—and even most suggestions—of how to use this approach to prove lower bounds find a closed set $\mathcal{D}_n$ that is disjoint from $\overline{\mathcal{C}_n}$, putting them back into the framework of approach (1). Algebraic pseudorandom generators are very close to hitting sets, and all known construction of hitting sets for special cases of PIT in fact work against the Zariski closure of the sets considered. This trend of hitting sets working against Zariski closures of complexity classes goes back to the first (probabilistic)*

[20] Similar ideas can be traced back to Heintz & Schnorr (1982), who showed by a similar proof that from a hitting set one can explicitly construct a polynomial that requires large algebraic circuits to compute (compare their Theorem 4.5 to Agrawal's Theorem 51). Of course, Heintz and Schnorr published their paper before the idea of pseudorandom generator was introduced into Boolean complexity (Blum & Micali 1984; Nisan & Wigderson 1994; Yao 1982) and became a mainstay of research in complexity theory.

construction of a hitting set due to Heintz & Schnorr (1982) *and was recently reiterated and extended by* Mulmuley (2012).

Each of these three approaches of course requires some insight: (1) requires finding a test polynomial with the desired properties, (2) requires finding all test polynomials that vanish on $\mathcal{B}_n$ (or at least a generating set[21] for the ideal of such polynomials), and (3) requires finding the set $\mathcal{D}_n$ along with all the test polynomials that vanish on $\mathcal{D}_n$ (or a generating set thereof).

However, barring some miraculous leap of ingenuity—which of course we cannot rule out—we can compare the a priori difficulty of these approaches:

(1) requires finding a single test polynomial t, verifying that t vanishes on $\mathcal{C}_n$ (which implies that it vanishes on $\overline{\mathcal{C}_n} = \mathcal{A}_n$), and verifying that $t(f_{hard,n}) \neq 0$.

(2) requires finding or knowing a generating set[21] $t_1, \dots, t_k$ for the set of test polynomials that vanish on $\mathcal{B}_n$ and then verifying that $t_i(f_{hard,n}) = 0$ for all $1 \leq i \leq k$.

(3) requires constructing $\mathcal{D}_n$, along with a generating set[21] t_1, $\dots$, t_k for the set of test polynomials that vanish on it, verifying that $t_i(f_{hard,n}) = 0$ for all $1 \leq i \leq k$, and verifying that $\mathcal{D}_n$ is disjoint from $\mathcal{C}_n$.

First, there is the obvious difference that (1) only requires finding a *single* polynomial and verifying its properties, whereas both (2) and (3) require finding a whole set of polynomials and verifying their properties. Furthermore, in most such situations the number of polynomials needed in (2) and (3) will be exponential in n: In all the examples, we are aware of except for Remark B.10, the sets $\mathcal{A}_n, \mathcal{B}_n, \mathcal{C}_n, \mathcal{D}_n$ have dimension poly(n) and live in a space like Poly$^{O(n)}(x_1, \dots, x_n)$ of dimension $2^{O(n \log n)}$, which implies that

[21] In fact, we do not even need a generating set $\{t_i\}$, we just need that $\{x : t_1(x) = \cdots = t_k(x) = 0\} = \mathcal{B}_n$, or equivalently that the t_i generate an ideal whose radical is the ideal of all functions that vanish on $\mathcal{B}_n$. We call such a set a set of *defining equations* for $\mathcal{B}_n$. The lower bound we give on the required number of such t_i holds for both generating sets and defining sets.

any generating set (or even a defining set[21]) will require roughly $2^{O(n \log n)}$ generators.[22]

Remark B.10. *In the case of $n \times n$ matrix multiplication, the ambient space has dimension n^6, and in the case of matrix rigidity, the ambient space has dimension n^2, so the preceding point about dimensions is not an issue. However, it may be telling that even in these cases, the approach via test polynomials seems to be the most successful so far. In the case of matrix multiplication, this corresponds to border rank, which has been successfully used for upper bounds* (Coppersmith & Winograd 1990; Schönhage 1981; Stothers 2010; Vassilevska Williams 2012) *as well as lower bounds* (Bläser 1999; Bürgisser & Ikenmeyer 2013; Hauenstein, Ikenmeyer & Landsberg 2013; Landsberg 2014b; Landsberg & Ottaviani 2011; Massarenti & Raviolo 2012; Strassen 1983) *(see* Example B.1*). In the case of matrix rigidity, see* Gesmundo, Hauenstein, Ikenmeyer & Landsberg (2013); Kumar, Lokam, Patankar & Sarma (2009); Landsberg, Taylor & Vishnoi (2003)*, the last of which is the most recent and uses separating modules as well.*

Second, we can use the complexity of the corresponding verification problems as a guide to the mathematical difficulty of the associated proofs. In order to compare these approaches on fair ground, we will evaluate their complexity relative to the complexity of evaluating a given test polynomial t at a given f (in practice this step may already be difficult for asymptotic results). We thus assume that test polynomials are given by algebraic circuits, and then, we measure the complexity of the problem relative to the size of such circuits.

1. Verifying that t vanishes on $\mathcal{C}_n$ can be reduced to an instance of polynomial identity testing and is thus in coRP: If $\mathcal{C}_n$

[22] Note that, in many situations, B_n has very small codimension in A_n, even codimension 1, as in Kumar (2013). This would mean that, if we knew that $f_{hard,n}$ were in A_n, we would only need to find a single polynomial t that cuts B_n out of A_n and then verify that $t(f_{hard,n}) = 0$. However, as we are explicitly not assuming in approach (2) that $f_{hard,n} \in A_n$—let alone that one knows this fact—showing that $f_{hard,n}$ is in B_n seems to require checking exponentially many test polynomials.

is the image of a simple algebraic map φ from some $\mathbb{F}^N$, as most complexity classes are (see the previous section), we can generate random points of $\mathcal{C}_n$ by applying φ to random points in $\mathbb{F}^N$ and then evaluate t on the result. In all situations, we are aware of $N \leq \text{poly}(n)$.

2. Verifying that $f_{hard,n} \in \mathcal{B}_n$ requires knowing a generating set[21] of the test polynomials that vanish on $\mathcal{B}_n$, which as we argued above will typically consists of exponentially many test polynomials, and hence require at least that much time to verify.

3. Even if $\mathcal{D}_n$ is chosen to have a small generating set[21], avoiding the difficulty of (2), verifying that $\mathcal{D}_n$ is disjoint from $\mathcal{C}_n = \text{Im}(\varphi_n)$ reduces to deciding whether a variety given by equations—namely, the equations $t_i(\varphi(\mathbf{x})) = 0$ for $1 \leq i \leq k$, which define $\varphi^{-1}(\mathcal{D}_n)$—is empty or not, which is NP-hard in general. We refer to this problem as the computational problem of Hilbert's Nullstellensatz HN. Furthermore, as the φ_n are very simple, if we treat the generators $t_1, \ldots, t_k$ as the input to our verification problem, the verification problem here is likely to be just as general as HN.

Also note that the fewer generators there are for the set of test polynomials that vanish on $\mathcal{D}_n$, the larger $\mathcal{D}_n$ is, and hence, the less likely it is to be disjoint from $\mathcal{C}_n$. This makes it seem unlikely that one could in fact find a $\mathcal{D}_n$ described by few test polynomials, let alone that the corresponding instance of HN would not be a hard instance. Either way, we find the fact that the verification problem for (1) is in coRP, the verification problem for (2) seems to take exponential time, and the verification problem for (3) is NP-hard very suggestive.

Finally, in the absence of a brilliant insight to construct a $\mathcal{D}_n$ that has exponential dimension and yet is both disjoint from $\mathcal{C}_n$ and avoids the difficulty of HN, the ease of verification in (1) suggests that a *feasible computational approach* is possible using a brute force search for test modules, whereas this is not the case for approaches (2) and (3). We do not expect such an approach to resolve asymptotic complexity questions on its own, but it may be a useful tool.

To see, at a high level, why searching for separating modules can make the search for lower bounds more feasible, let us contrast with a naive computer search. To show a Boolean lower bound of size 25 on some function of 10 variables, one might be able to use modern SAT solvers to search over all 25-gate circuits on a standard desktop, but already at these small numbers this would be pushing the limits of modern computation. Even today's largest super-computers are unlikely to be able to prove a lower bound of size 100 using state-of-the-art SAT solvers. Furthermore, such concrete lower bounds may not shed much light on the overall asymptotic circuit complexity of the given problem.

In contrast, the search for separating modules for finite-size lower bounds is aided by several factors. First, efficient-in-practice algorithms for certain representation-theoretic multiplicities of the symmetric groups [Kronecker coefficients, implemented in many standard computer algebra systems such as GAP (GAP 2014) and MAGMA (Bosma *et al.* 1997)] can be used to rule out many test modules without having to evaluate any test polynomials. Second, given a test polynomial explicitly as, say, an algebraic circuit, determining whether it evaluates to zero or not is easy by evaluating it at random points in the class, as above. Although this need not be trivially easy in practice, it is almost certainly easier than the SAT solver approach. Hauenstein, Ikenmeyer & Landsberg (2013) have demonstrated how to use computers efficiently to search for new separating modules; indeed, they found a separating module that gave a new, simpler proof of Landsberg's major result that the border rank of 2 × 2 matrix multiplication is exactly 7 (Landsberg 2006). Finally, and perhaps most importantly, finite-sized lower bounds found via separating modules are more likely to generalize to yield asymptotic lower bounds. Indeed, this is essentially how Bürgisser & Ikenmeyer (2011); Bürgisser & Ikenmeyer (2013) proved their recent lower bounds on matrix multiplication. Since separating modules are group representations, and representations of S_n and GL_n naturally come in infinite families, generalizing a separating module to an infinite family of separating modules is often direct. Although one then requires a proof that the test modules in this infinite family are indeed separating, at least one has some idea of which test modules to look at. Either way, the extension to an asymptotic result is certainly much easier than proving an asymptotic result with "inspiration" from a circuit-size lower bound of size 100 found via SAT solvers.

The preceding discussion suggests two questions of interest:

Open Question B.11. *Is the variant of* HN *above still* NP*-hard? More specifically, fix a polynomial map* φ*, and consider the problem* HN_φ*: given polynomials* $t_1, \ldots, t_k$*, decide whether the variety defined by* $t_1(\varphi(\vec{x})) = \cdots = t_k(\varphi(\vec{x})) = 0$ *is empty. For which* φ *is* HN_φ NP*-hard?*

Given a Zariski-constructible set $\mathcal{C}$, it can be written as some "Boolean" combination of Zariski-closed sets, that is, using unions, intersections, and complements. Consider this Boolean combination as a circuit in its own right, with union, intersection, and complement gates, whose inputs correspond to Zariski-closed sets. The representation of a constructible set as a union of locally closed sets roughly corresponds to depth 2 Boolean combinations. The complexity of $\mathcal{C}$ as a constructible set can be measured in terms of both the complexities of the Zariski-closed inputs to the Boolean combination, and in terms of the complexity of the Boolean combination itself.

Open Question B.12. *Given upper bounds on the degree, circuit size, and/or monomial sparsity of a polynomial map* φ*, what can we say about the complexity of the image of* φ*, in terms of how many Zariski-closed sets are needed to describe it, the complexity of the ideals of those Zariski-closed sets, and the complexity of the Boolean combination as above?*

Presumably, there is some trade-off between these measures: To some extent, one may be able to simplify the Boolean combination used by using more Zariski-closed sets or Zariski-closed sets that are more complicated.

B.4. The utility of separating modules: geometric aspects. Two aspects of the boundary points of border-$\mathcal{C}$ suggest that considering $\mathcal{C}$ without considering border-$\mathcal{C}$ amounts to little more than sweeping essential difficulties under the rug, namely (1) the presence of functions in $\overline{\mathcal{C}}_n$ that may not even be reducible to functions in $\mathcal{C}_{\text{poly}(n)}$; and (2) non-normality (in the technical, algebro-geometric sense, which we explain below). To make our discussion here somewhat more concrete, we consider the class of functions $\mathcal{D}et$ of polynomial border-determinantal complexity; that is, $\mathcal{D}et_n$ consists of all limits of projections of the $n \times n$ determinant. It turns out that $\mathcal{D}et_n$ is in fact an orbit closure and is currently perhaps the most well-studied orbit closure in the GCT Program.

Complexity versus border complexity. The notion of *order of approximation* (see, e.g., Bürgisser 2004; Bürgisser, Landsberg, Manivel & Weyman 2011b; Grochow 2012) captures quantitatively how quickly a point in the boundary of $\overline{\mathcal{C}}_n$ can be approached by points in $\mathcal{C}_n$. In the setting of matrix multiplication, considering the order of approximation together with the tensor power trick allowed Bini to show that the exponent of matrix multiplication is the same whether calculated with tensor rank or border rank (Bini 1980). In the context of permanent versus determinant, Bürgisser *et al.* (2011b, Proposition 9.4.3) and the author (Grochow 2012, Proposition 3.5.2) independently showed that functions in $\mathcal{D}et_n$ with order of approximation poly(n) are in fact projections of only a polynomially larger determinant. However, the best known upper bound for the order of approximation is exponential. The difference between showing that the $n \times n$ permanent perm_n is not in $\mathcal{D}et_{\text{poly}(n)}$ and showing that perm_n has super-polynomial order of approximation is analogous to the difference between a non-computability result and a complexity lower bound (the latter generally being much harder). In this case, although both statements—$\text{perm}_n \notin \mathcal{D}et_{\text{poly}(n)}$ and the super-polynomial order of approximation of perm_n within $\mathcal{D}et_{\text{poly}(n)}$—are quantitative statements, the latter is much more refined than the former, involving two different asymptotic quantities to be estimated rather than just one. The possible existence of functions of polynomial border-determinantal complexity but super-polynomial deter-

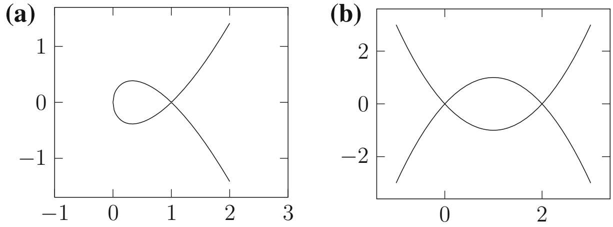

Figure B.1: Examples of non-normal varieties. **a** The nodal cubic $y^2 = x(x-1)^2$ is not normal at the point $(1, 0)$. **b** The union of two parabolas, defined by $(y+1-(x-1)^2)(y-1+(x-1)^2) = 0$, is not normal at the two intersection points $(0, 0)$ and $(0, 2)$.

minantal complexity makes a potential lower bound (at least seem) significantly more complicated, yet without considering border-$\mathcal{C}$ we would never have even been aware of this fundamental issue which deserves further study.

We note that Mulmuley and Sohoni constructed a function that they conjecture to have this intermediate status: polynomial border-determinantal complexity but super-polynomial determinantal complexity (Mulmuley & Sohoni 2001, Section 4.2).

Non-normality (intuitive picture). Normality is a "niceness" property of algebraic varieties that is akin to smoothness (indeed, smooth varieties are always normal), but is flexible enough to allow some singularities (see, e.g., Eisenbud (1995, Section 4.2) or Shafarevich (1994, Section I.5) for introductory treatments of normality, and the books Greco (1978) or Vasconcelos (2005) for more in-depth treatments). The definition of normality, though not difficult to state, is somewhat far from any geometric intuition, so here we start here by giving a few pictures, so that we can discuss at an intuitive level what normality tells us about (border-)complexity.

Normality is in fact a local property like smoothness, in that there is a notion of a point v being normal in an algebraic set V, and V is then normal if and only if it is normal at every point. For example, the varieties in Figure B.1 are not normal, but only at the points of (self-)intersection.

One geometric interpretation of normality is that there is "no undue gluing of subvarieties or tangent spaces" (Schwede 2012).

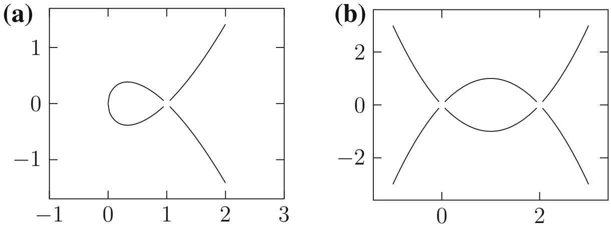

Figure B.2: Considering an open dense subset of non-normal varieties leaves something to be desired. **a** The nodal cubic with its non-normal point removed. **b** The union of two parabolas, with its non-normal points removed.

Explaining this in detail requires the language of schemes, but we can at least say where the undue gluing is in Figure B.1. Figure B.1a can essentially be obtained from the real line by identifying the points 0 and 1 (and then rotating)—an "undue gluing." Figure B.1b can obviously be obtained from the *disjoint* union of two parabolas by gluing them together at two points.

Kumar (2013) was, to our knowledge, the first to show that the closure of a natural complexity class was not normal; he showed this for projections of the determinant and for projections of the permanent. The same conclusion holds for border rank in the context of matrix multiplication, and for depth 3 algebraic border circuits (Landsberg 2014a, Section 9), we (thus) expect that this phenomenon holds more generally for many complexity classes. The fact that normality does not hold when including the boundary starts to suggest that the "geometric complexity" of the boundary $\overline{\mathcal{C}}\backslash\mathcal{C}$ gives some measure of the difficulty of proving lower bounds against $\mathcal{C}$. We give further evidence for this below.

When border-$\mathcal{C}$ is not normal, considering only $\mathcal{C}$ without considering border-$\mathcal{C}$ is like looking at Figure B.2 instead of Figure B.1.

In Figure B.2, the non-normal points are gone and what is left is a constructible, normal (even smooth) set, yet we see visually that something is missing. Indeed, the gestalt phenomenon in the human visual system makes it almost impossible to look at Figure B.2 without thinking that the corresponding lines actually do intersect, perhaps behind a small white dot. The sets still "want" to intersect themselves in non-normal ways, it is just we are ignoring these non-normal intersections by fiat. Although our visual intuition fails us for complexity classes, because they are much higher-dimensional objects, the lesson is the same: The complexity classes "want" to bend and fold in on themselves in non-normal ways, sometimes on the boundary. Perhaps not considering border complexity does not make our lives easier by smoothing things out, but only makes us blind to fundamental geometric phenomena that, if we could see in exponentially many dimensions, would be right before our eyes.

Non-normality (technical argument). Had normality held, it would have been a very useful property, as we now explain.

As already mentioned, any class with a problem that is complete under projections, as in Corollary B.6(ii), is in fact an orbit closure. More precisely, if $(f_n(\vec{x}))_{n=1}^{\infty}$ is a complete family (presumably m here grows at most polynomially with n), then one may take $\mathcal{C}_n$ to be $\{f_n(Ax) | A \in M_m(\mathbb{F})\}$. Since the invertible linear transformations of the variables are dense in $M_m(\mathbb{F})$, the alternative $\mathcal{C}'_n = \{f_n(Ax) | A \in \mathrm{GL}_m(\mathbb{F})\}$ has the same closure: $\overline{\mathcal{C}_n} = \overline{\mathcal{C}'_n}$. The advantage of considering $\mathcal{C}'_n$ instead of $\mathcal{C}_n$ is that $\mathcal{C}'_n$ is a single orbit of the algebraic group GL_m, and much can be said about the geometry and representation theory of this orbit.

In particular, if one considers not merely test polynomials, but test *rational functions* (ratios of polynomials), then much of the discussion in this paper about how to use separating modules goes through with some additional caveats. By a *rational test module* we mean a test module that consists of test rational functions. Although rational functions in some ways introduce additional complications—which will ultimately be the source of the difficulty that non-normality poses—they also simplify certain aspects of the problem. For example, if we let $\tilde{m}_\lambda(\mathcal{C}'_n)$ denote the multiplicity of rational test modules of type λ that are supported on $\mathcal{C}'_n$, then **there is a clean mathematical way, at least in principle, to calculate the** $\tilde{m}_\lambda(\mathcal{C}'_n)$.[23]

[23] Namely, if one takes the complete function f_n as above and considers its stabilizer $\mathrm{Stab}(f_n)$ in $\mathrm{GL}_m(\mathbb{F})$, the ring of rational functions that are supported on the orbit $\mathcal{C}'_n$ is simply the ring of rational functions on $\mathrm{GL}_m(\mathbb{F})$ that are invariant under $\mathrm{Stab}(f_n)$ (see, e.g., Goodman & Wallach 2009, Theorem 12.1.4).

The question then becomes: Which rational test modules supported on the orbit $\mathcal{C}'_n$ can be extended to rational test modules supported on the closure $\overline{\mathcal{C}'_n}$? By a standard result in algebraic geometry, a rational function that is defined everywhere on a closed set is in fact a polynomial, so a rational test module that is supported on the closure $\overline{\mathcal{C}'_n}$ is in fact a polynomial test module.

Now that we know what we are after, we give a definition equivalent to normality:

Definition & Theorem B.13 (Serre, see, e.g., Eisenbud 1995, Theorem 11.5). *A variety V of dimension d is normal if and only if*

(i) Its set of singular points has dimension at most $d-2$; and

(ii) Rational functions on V essentially satisfy Hartogs's Extension Theorem: If f is a ratio of two polynomials on V, and the subset of V where f is not defined has dimension at most $d-2$, then f uniquely extends to a polynomial function on V.

One can imagine how useful condition (2) would be in relating $\tilde{m}_\lambda(\mathcal{C}'_n)$ and $m_\lambda(\overline{\mathcal{C}'_n})$. This puts some real technical muscle behind the idea that the "geometric complexity" of the boundary of a complexity class gives some measure of the difficulty of proving lower bounds against that class. Despite perhaps being unintuitive at first, based on our current knowledge it seems that understanding the boundary of a complexity class is a key step toward understanding the class itself.

C. Discussion of terminology

The new terminology we introduced in this paper was far from arbitrary; here we explain our reasons for choosing the terminology we did. A test GL_n-module is, in particular, a representation of GL_n. Indeed, the word "module" is often used interchangeably with "representation" in representation theory. In our setting, it

Footnote 23 continued
The question of which GL_m representations contain a $\mathrm{Stab}(f_n)$-invariant is by no means trivial, but at least conceptually it is quite clean.

has the additional connotation of a "module of tests" in the sense of computer programming. We believe the phrase "test module" is new.

Separating $\mathrm{GL}_n(\mathbb{C})$-modules are essentially equivalent to the "HWV obstructions" of Bürgisser & Ikenmeyer (2013). In particular, the smallest $\mathrm{GL}_n(\mathbb{C})$-module containing an HWV obstruction is a separating module, and every separating $\mathrm{GL}_n(\mathbb{C})$-module contains some HWV obstruction (see Bürgisser & Ikenmeyer 2013, Proposition 3.3). We use our terminology as it generalizes (see Section 2.4) to other groups for which the highest weight theory does not apply, and we believe it is simpler to understand for expository purposes—in particular, it does not require knowing anything about Lie theory and the theory of highest weights. However, for certain approaches to certain lower bounds there are technical advantages to considering the highest weight vectors directly, as in Bürgisser & Ikenmeyer (2013).

D. Standard notation in the literature

Rather than $\mathrm{Poly}^d(x_1, \ldots, x_n)$, it is standard to see $\mathrm{Sym}^d(\mathbb{C}^n)$, $\mathrm{Sym}^d(\mathbb{C}^{n*})$, $S^d(\mathbb{C}^n)$, or $S^d(\mathbb{C}^{n*})$, or even $\mathrm{Sym}^d(V)$ or $\mathrm{Sym}^d(V^*)$, or any other combination of these notations. The use of $\mathbb{C}^{n*}$ or V^* here comes from a viewpoint in which the variables x_i are viewed as the coordinate functions on an n-dimensional vector space $V = \mathbb{C}^n$, hence are elements of its (linear) dual vector space $V^* = \mathbb{C}^{n*}$. Sometimes the dual is dropped because it does not affect many statements. The use of Sym^d or S^d is to denote the "symmetric product" to distinguish it from, say, the tensor product (which corresponds to non-commutative polynomials) or the wedge product (which corresponds to anti-commutative tensors, for which $x_i x_j = -x_j x_i$).

The space of test polynomials of degree D is then denoted $\mathrm{Sym}^D(\mathrm{Sym}^d(\mathbb{C}^n))$ (or variations similar to the above). Continuing with the viewpoint above, the coefficients a_e of a polynomial $f \in \mathrm{Sym}^d(\mathbb{C}^{n*})$ are viewed as linear functions on the space of input polynomials, hence as elements of the dual vector space $\mathrm{Sym}^d(\mathbb{C}^n)$. Polynomials in the a_e then live in the D-th symmetric power, as before.

The space of test polynomials is sometimes denoted

$$\mathbb{C}\left[\mathrm{Sym}^d(\mathbb{C}^{n*})\right] \qquad \text{or} \qquad \mathcal{O}\left(\mathrm{Sym}^d(\mathbb{C}^{n*})\right).$$

These are standard notations in algebraic geometry for the coordinate ring of the affine algebraic variety $\mathrm{Sym}^d(\mathbb{C}^n)$.

A GL_n-module of type λ is typically referred to as a Weyl module, which has several more-or-less standard notations: V_λ, $V_\lambda(\mathrm{GL}_n)$, $\mathbb{S}_\lambda(V)$ when the group is $\mathrm{GL}(V)$ ("$\mathbb{S}$" for "Schur functor"), or $\{\lambda\}$.

An S_n-module of type λ is typically referred to as a Specht module, which also has several more-or-less standard notations, including S_λ and $[\lambda]$.

In both the above cases, λ typically refers to a partition, as the irreducible modules of $\mathrm{GL}_n(\mathbb{C})$ are in bijective correspondence with partitions with at most n parts, and the irreducible modules of S_n over $\mathbb{C}$ are in bijective correspondence with partitions of the number n.

References

Scott Aaronson & Andrew Drucker (2009). Impagliazzo's worlds in arithmetic complexity. Talk presented at the Workshop on Complexity and Cryptography: Status of Impagliazzo's Worlds, Center for Computational Intractability, Princeton, NJ, June 5, 2009. Slides available at http://www.scottaaronson.com/talks/arith.ppt.

B. Adsul, Milind Sohoni & K. V. Subrahmanyam (2009). Quantum deformations of the restriction of $GL_{mn}(\mathbb{C})$-modules to $GL_m(\mathbb{C}) \times GL_n(\mathbb{C})$. arXiv:0905.0094 [math.RT].

Manindra Agrawal (2005). Proving lower bounds via pseudo-random generators. In *FSTTCS 2005: Foundations of software technology and theoretical computer science*, volume 3821 of *Lecture Notes in Computer Science*, 92–105. Springer, Berlin.

Manindra Agrawal & V. Vinay (2008). Arithmetic Circuits: A Chasm at Depth Four. In *FOCS '08: 49th Annual IEEE Symposium on Foundations of Computer Science*, 67–75. IEEE Computer Society.

ERIC ALLENDER (1999). The permanent requires large uniform threshold circuits. *Chicago J. Theoret. Comput. Sci.* Art. 7.

ERIC ALLENDER & VIVEK GORE (1994). A uniform circuit lower bound for the permanent. *SIAM J. Comput* **23**(5), 1026–1049.

SAUGATA BASU, RICHARD POLLACK & MARIE-FRANÇOISE ROY (2006). *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2nd edition.

WALTER BAUR & VOLKER STRASSEN (1983). The complexity of partial derivatives. *Theoret. Comput. Sci.* **22**(3), 317–330.

MICHAEL BEN-OR (1983). Lower bounds for algebraic computation trees. In *STOC '83: 15th Annual ACM Symposium on Theory of Computing*, 80–86. ACM.

DARIO BINI (1980). Relations between exact and approximate bilinear algorithms. Applications. *Calcolo* **17**(1), 87–97.

DARIO BINI, MILVIO CAPOVANI, FRANCESCO ROMANI & GRAZIA LOTTI (1979). $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inform. Process. Lett.* **8**(5), 234–235.

ANDERS BJÖRNER, LÁSZLÓ LOVÁSZ & ANDREW C. C. YAO (1992). Linear decision trees: volume estimates and topological bounds. In *STOC '92: 24th Annual ACM Symposium on Theory of Computing*, 170–177. ACM.

MARKUS BLÄSER (1999). A $\frac{5}{2}n^2$-lower bound for the rank of $n \times n$-matrix multiplication over arbitrary fields. In *FOCS '99: 40th Annual IEEE Symposium on Foundations of Computer Science*, 45–50. IEEE Computer Soc., Los Alamitos, CA.

JONAH BLASIAK (2012). Kronecker coefficients for one hook shape. arXiv:1209.2018 [math.CO].

JONAH BLASIAK, KETAN D. MULMULEY & MILIND SOHONI (2015). Geometric complexity theory IV: nonstandard quantum group for the Kronecker problem. *Mem. Amer. Math. Soc.* **235**(1109).

MANUEL BLUM & SILVIO MICALI (1984). How to generate cryptographically strong sequences of pseudorandom bits. *SIAM J. Comput* **13**(4), 850–864.

Jacek Bochnak, Michel Coste & Marie-Françoise Roy (1998). *Real algebraic geometry*, volume 36 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3)*. Springer-Verlag, Berlin.

Wieb Bosma, John Cannon & Catherine Playoust (1997). The Magma algebra system. I. The user language. *J. Symbolic Comput.* **24**(3-4), 235–265. Computational algebra and number theory (London, 1993).

Michel Brion (2011). Invariant Hilbert schemes. arXiv:1102.0198v2 [math.AG].

Harry Buhrman, Lance Fortnow & Thomas Thierauf (1998). Nonrelativizing separations. In *CCC '98: 13th IEEE Conference on Computational Complexity*, 8–12. IEEE Computer Soc., Los Alamitos, CA.

Peter Bürgisser (2004). The complexity of factors of multivariate polynomials. *Found. Comput. Math.* **4**(4), 369–396.

Peter Bürgisser, Matthias Christandl & Christian Ikenmeyer (2011a). Nonvanishing of Kronecker coefficients for rectangular shapes. *Adv. Math.* **227**(5), 2082–2091.

Peter Bürgisser, Michael Clausen & M. Amin Shokrollahi (1997). *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, Berlin.

Peter Bürgisser & Christian Ikenmeyer (2009). A max-flow algorithm for positivity of Littlewood–Richardson coefficients. In *21st Annual International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2009)*, Discrete Math. Theor. Comput. Sci. Proc., AJ. Assoc. Discrete Math. Theor. Comput. Sci., Nancy.

Peter Bürgisser & Christian Ikenmeyer (2011). Geometric complexity theory and tensor rank. In *STOC '11: 43rd Annual ACM Symposium on Theory of Computing*, 509–518. ACM.

Peter Bürgisser & Christian Ikenmeyer (2013). Deciding Positivity of Littlewood–Richardson Coefficients. *SIAM J. Discrete Math* **27**, 1639–1681.

PETER BÜRGISSER & CHRISTIAN IKENMEYER (2013). Explicit lower bounds via geometric complexity theory. In *STOC '13: 45th Annual ACM Symposium on Theory of Computing*, 141–150. ACM, New York.

PETER BÜRGISSER, J. M. LANDSBERG, LAURENT MANIVEL & JERZY WEYMAN (2011b). An overview of mathematical issues arising in the Geometric Complexity Theory approach to $\mathsf{VP} \neq \mathsf{VNP}$. *SIAM J. Comput* **40**(4), 1179–1209.

SAMUEL BUSS & RYAN WILLIAMS (2012). Limits on Alternation-Trading Proofs for Time-Space Lower Bounds. In *CCC '12: 27th IEEE Conference on Computational Complexity*, 181–191. IEEE Computer Society.

PATRICIA J. CARSTENSEN (1983a). Complexity of some parametric integer and network programming problems. *Math. Programming* **26**(1), 64–75.

PATRICIA JUNE CARSTENSEN (1983b). *The complexity of some problems in parametric linear and combinatorial programming.* Ph.D. thesis, University of Michigan—Ann Arbor, Ann Arbor, MI.

CLAUDE CHEVALLEY & HENRI CARTAN (1955–1956). Schémas normaux; morphismes; ensembles constructibles. In *Séminaire Henri Cartan, tome 8*, Exp. No. 7, 1–10.

STEPHEN A. COOK (1973). A hierarchy for nondeterministic time complexity. *J. Comput. System Sci.* **7**, 343–353. STOC '72: 4th Annual ACM Symposium on Theory of Computing.

DON COPPERSMITH & SHMUEL WINOGRAD (1990). Matrix multiplication via arithmetic progressions. *J. Symbolic Comput.* **9**(3), 251–280.

DAVID COX, JOHN LITTLE & DONAL O'SHEA (1997). *Ideals, varieties, and algorithms.* Undergraduate Texts in Mathematics. Springer-Verlag, New York, 2nd edition.

V. I. DANILOV (1994). Algebraic varieties and schemes. In *Algebraic geometry, I*, volume 23 of *Encyclopaedia Math. Sci.*, 167–297. Springer, Berlin.

SCOTT DIEHL, DIETER VAN MELKEBEEK & RYAN WILLIAMS (2011). An improved time-space lower bound for tautologies. *J. Comb. Optim.* **22**(3), 325–338.

RODNEY G. DOWNEY & MICHAEL R. FELLOWS (1999). *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York.

DAVID EISENBUD (1995). *Commutative algebra*, volume 150 of *Graduate Texts in Mathematics*. Springer-Verlag, New York.

ISMOR FISCHER (1994). Sums of like powers of multivariate linear forms. *Math. Mag.* **67**(1), 59–61.

LANCE FORTNOW (2000). Time-space tradeoffs for satisfiability. *J. Comput. System Sci.* **60**(2, part 2), 337–353.

LANCE FORTNOW, RICHARD LIPTON, DIETER VAN MELKEBEEK & ANASTASIOS VIGLAS (2005). Time-space lower bounds for satisfiability. *J. Assoc. Comput. Mach.* **52**(6), 835–865.

WILLIAM FULTON & JOE HARRIS (1991). *Representation theory*, volume 129 of *Graduate Texts in Mathematics*. Springer-Verlag, New York.

GAP (2014). *GAP – Groups, Algorithms, and Programming, Version 4.7.6*. The GAP Group. URL http://www.gap-system.org.

FULVIO GESMUNDO, JONATHAN HAUENSTEIN, CHRISTIAN IKENMEYER & J. M. LANDSBERG (2013). Complexity of linear circuits and geometry. arXiv:1310.1362 [cs.CC].

ROE GOODMAN & NOLAN R. WALLACH (2009). *Symmetry, representations, and invariants*, volume 255 of *Graduate Texts in Mathematics*. Springer, Dordrecht.

SILVIO GRECO (1978). *Normal varieties*. Institutiones Mathematicae, IV. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London.

BERNHARD GRIESSER (1986). Lower bounds for the approximative complexity. *Theoret. Comput. Sci.* **46**(2-3), 329–338.

D. GRIGORIEV & A. RAZBOROV (2000). Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Appl. Algebra Engrg. Comm. Comput.* **10**(6), 465–487.

Dima Grigoriev & Marek Karpinski (1998). An exponential lower bound for depth 3 arithmetic circuits. In *STOC '98: 30th Annual ACM Symposium on Theory of Computing*, 577–582. ACM.

Joshua A. Grochow (2012). *Symmetry and Equivalence Relations in Classical and Geometric Complexity Theory.* Ph.D. thesis, University of Chicago, Chicago, IL.

Joshua A. Grochow (2014). Unifying known lower bounds via geometric complexity theory (extended abstract). In *CCC '14: 29th IEEE Conference on Computational Complexity.* IEEE.

Alexander Grothendieck (1964). Éléments de géométrie algébrique. IV. Étude locale des schémas et des morphismes de schémas. I. *Inst. Hautes Études Sci. Publ. Math.* (20), 259.

Alexander Grothendieck (1995). Techniques de construction et théorèmes d'existence en géométrie algébrique. IV. Les schémas de Hilbert. In *Séminaire Bourbaki, Vol. 6*, Exp. No. 221, 249–276. Soc. Math. France, Paris.

Ankit Gupta, Pritish Kamath, Neeraj Kayal & Ramprasad Saptharishi (2012). Approaching the chasm at depth four. Technical Report TR12-098, Electronic Colloquium on Computational Complexity.

Ankit Gupta, Pritish Kamath, Neeraj Kayal & Ramprasad Saptharishi (2013). Arithmetic circuits: A chasm at depth three. In *FOCS '13: 54th Annual IEEE Symposium on Foundations of Computer Science.*

Robert M. Hardt (1980). Semi-algebraic local-triviality in semi-algebraic mappings. *Amer. J. Math.* **102**(2), 291–302.

Juris Hartmanis & R. E. Stearns (1965). On the computational complexity of algorithms. *Trans. Amer. Math. Soc.* **117**, 285–306.

Jonathan D. Hauenstein, Christian Ikenmeyer & J. M. Landsberg (2013). Equations for lower bounds on border rank. *Exp. Math.* **22**(4), 372–383.

Joos Heintz & C.-P. Schnorr (1982). Testing polynomials which are easy to compute. In *Logic and algorithmic (Zurich, 1980)*, volume 30 of *Monograph. Enseign. Math.*, 237–254. Univ. Genève, Geneva.

Joos Heintz & Malte Sieveking (1980). Lower bounds for polynomials with algebraic coefficients. *Theoret. Comput. Sci.* **11**(3), 321–330.

Christian Ikenmeyer (2012). *Geometric complexity theory, tensor rank, and Littlewood–Richardson coefficients*. Ph.D. thesis, Univ. Paderborn. URL http://digital.ub.uni-paderborn.de/ubpb/urn/urn:nbn:de:hbz:466:2-10472.

Maurice Jansen & Rahul Santhanam (2012). Marginal hitting sets imply super-polynomial lower bounds for permanent. In *ITCS '12: 3rd Annual ACM Innovations in Theoretical Computer Science*, 496–506. ACM.

Maurice Jansen & Rahul Santhanam (2013). Permanent does not have succinct polynomial size arithmetic circuits of constant depth. *Inform. and Comput.* **222**, 195–207.

Ravi Kannan (1982). Circuit-size lower bounds and nonreducibility to sparse sets. *Inform. and Control* **55**(1-3), 40–56.

Neeraj Kayal (2011). Affine projections of polynomials. Technical Report TR11-061, Electronic Colloquium on Computational Complexity.

Pascal Koiran (2012). Arithmetic circuits: the chasm at depth four gets wider. *Theoret. Comput. Sci.* **448**, 56–65.

Pascal Koiran & Sylvain Perifel (2009). A superpolynomial lower bound on the size of uniform non-constant-depth threshold circuits for the permanent. In *CCC '09: 24th IEEE Conference on Computational Complexity*, 35–40. IEEE Computer Soc., Los Alamitos, CA.

Abhinav Kumar, Satyanarayana V. Lokam, Vijay M. Patankar & M.N. Jayalal Sarma (2009). Using Elimination Theory to construct Rigid Matrices. arXiv:0910.5301 [cs.CC].

Shrawan Kumar (2013). Geometry of orbits of permanents and determinants. *Comment. Math. Helv.* **88**(3), 759–788.

J. M. Landsberg (2006). The border rank of the multiplication of 2 × 2 matrices is seven. *J. Amer. Math. Soc.* **19**(2), 447–459.

J. M. Landsberg (2008). Geometry and the complexity of matrix multiplication. *Bull. Amer. Math. Soc. (N.S.)* **45**(2), 247–284.

J. M. Landsberg (2014a). Geometric complexity theory: an introduction for geometers. *Annali Dell'universitá di Ferrara* 1–53. Preprint available as arXiv:1305.7387 [math.AG].

J. M. Landsberg (2014b). New lower bounds for the rank of matrix multiplication. *SIAM J. Comput.* **43**(1), 144–149.

J. M. Landsberg & Giorgio Ottaviani (2011). New lower bounds for the border rank of matrix multiplication. arXiv:1112.6007 [cs.CC].

J. M. Landsberg, J. Taylor & N. K. Vishnoi (2003). The geometry of matrix rigidity. Technical Report GIT-CC-03-54, Georgia Institute of Technology.

Joseph M. Landsberg, Laurent Manivel & Nicolas Ressayre (2013). Hypersurfaces with degenerate duals and the geometric complexity theory program. *Comment. Math. Helv.* **88**(2), 469–484.

Hông Vân Lê (2010). Constructing elusive functions with the help of evaluation mappings. arXiv:1011.2887v5 [math.LO].

Hông Vân Lê (2013). Lower bounds for the circuit size of partially homogeneous polynomials. arXiv:1302.3360 [cs.CC].

Guillaume Malod & Natacha Portier (2008). Characterizing Valiant's algebraic complexity classes. *J. Complexity* **24**(1), 16–38.

Alex Massarenti & Emanuele Raviolo (2012). The Rank of $n \times n$ Matrix Multiplication is at least $3n^2 - 2\sqrt{2}n^3/2 - 3n$. arXiv:1211.6320 [cs.CC].

Hideyuki Matsumura (1980). *Commutative algebra*, volume 56 of *Mathematics Lecture Note Series*. Benjamin/Cummings Publishing Co., Inc., Reading, Mass., 2nd edition.

Thierry Mignon & Nicolas Ressayre (2004). A quadratic bound for the determinant and permanent problem. *Int. Math. Res. Not.* (79), 4241–4253.

Ketan Mulmuley (1999). Lower bounds in a parallel model without bit operations. *SIAM J. Comput* **28**(4), 1460–1509 (electronic).

Ketan D. Mulmuley (2010). Explicit proofs and the flip. Technical report, Computer Science Department, The University of Chicago. URL http://gct.cs.uchicago.edu/gctflip.ps.

Ketan D. Mulmuley (2011a). Geometric Complexity Theory VI: the flip via positivity. Technical report, University of Chicago.

Ketan D. Mulmuley (2011b). On P vs. NP and Geometric Complexity Theory. *J. Assoc. Comput. Mach.* **58**(2), Art. 5.

Ketan D. Mulmuley (2012). Geometric Complexity Theory V: Equivalence between Blackbox Derandomization of Polynomial Identity Testing and Derandomization of Noether's Normalization Lemma. In *FOCS '12: 53rd Annual IEEE Symposium on Foundations of Computer Science*, 629–638.

Ketan D. Mulmuley & Milind Sohoni (2001). Geometric complexity theory I: an approach to the P vs. NP and related problems. *SIAM J. Comput* **31**(2), 496–526.

Ketan D. Mulmuley & Milind Sohoni (2008). Geometric complexity theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput* **38**(3), 1175–1206.

David Mumford (1976). *Algebraic geometry I. Complex projective varieties*, volume 221 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, Berlin.

Katta G. Murty (1980). Computational complexity of parametric linear programming. *Math. Programming* **19**(2), 213–219.

Noam Nisan & Avi Wigderson (1994). Hardness vs. randomness. *J. Comput. System Sci.* **49**(2), 149–167.

Noam Nisan & Avi Wigderson (1996/97). Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complexity* **6**(3), 217–234.

Igor Pak & Greta Panova (2014). On the complexity of computing Kronecker coefficients. To appear in *Computational Complexity*. arXiv:1404.0653.

Ran Raz (2006). Separation of multilinear circuit and formula size. *Theory Comput.* **2**, 121–135.

Ran Raz (2009). Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. Assoc. Comput. Mach.* **56**(2), Art. 8, 17.

Ran Raz (2010a). Elusive functions and lower bounds for arithmetic circuits. *Theory Comput.* **6**, 135–177.

Ran Raz (2010b). Tensor-rank and lower bounds for arithmetic formulas. In *STOC '10: 42nd Annual ACM Symposium on Theory of Computing*, 659–666. ACM.

Ran Raz, Amir Shpilka & Amir Yehudayoff (2008). A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. Comput* **38**(4), 1624–1647.

Ran Raz & Amir Yehudayoff (2009). Lower bounds and separations for constant depth multilinear circuits. *Comput. Complexity* **18**(2), 171–207.

Alexander A. Razborov (1987). Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Mat. Zametki* **41**(4), 598–607, 623. English translation: Mathematical Notes of the Academy of Sci. of the USSR, 41(4):333–338, 1987.

Alexander A. Razborov (1995a). Bounded arithmetic and lower bounds in Boolean complexity. In *Feasible mathematics, II (Ithaca, NY, 1992)*, volume 13 of *Progr. Comput. Sci. Appl. Logic*, 344–386. Birkhäuser Boston, Boston, MA.

Alexander A. Razborov (1995b). Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izv. Ross. Akad. Nauk Ser. Mat.* **59**(1), 201–224. English translation: Izvestiya. Mathematics. 59(1):205–227, 1995.

Alexander A. Razborov & Steven Rudich (1997). Natural proofs. *J. Comput. System Sci.* **55**(1, part 1), 24–35.

A. Schönhage (1981). Partial and total matrix multiplication. *SIAM J. Comput* **10**(3), 434–455.

Karl Schwede (2012). Answer to "Is there a "geometric" intuition underlying the notion of normal varieties?". Math Overflow, http://mathoverflow.net/a/109486/38434.

Igor R. Shafarevich (1994). *Basic algebraic geometry. 1.* Springer-Verlag, Berlin, 2nd edition.

Claude E. Shannon (1949). The synthesis of two-terminal switching circuits. *Bell System Tech. J.* **28**, 59–98.

Amir Shpilka & Avi Wigderson (2001). Depth-3 arithmetic circuits over fields of characteristic zero. *Comput. Complexity* **10**(1), 1–27.

Roman Smolensky (1987). Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *STOC '87: 19th Annual ACM Symposium on Theory of Computing*, 77–82. ACM.

Andrew Stothers (2010). *On the complexity of matrix multiplication.* Ph.D. thesis, University of Edinburgh, Edinburgh, UK.

Volker Strassen (1972/73). Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten. *Numer. Math.* **20**, 238–251.

Volker Strassen (1974). Polynomials with rational coefficients which are hard to compute. *SIAM J. Comput* **3**, 128–149.

Volker Strassen (1983). Rank and optimal computation of generic tensors. *Linear Algebra Appl.* **52/53**, 645–685.

Volker Strassen (1987). Relative bilinear complexity and matrix multiplication. *J. Reine Angew. Math.* **375/376**, 406–443.

Alfred Tarski (1948). *A Decision Method for Elementary Algebra and Geometry.* RAND Corporation, Santa Monica, Calif.

Sébastien Tavenas (2013). Improved Bounds for Reduction to Depth 4 and Depth 3. In *MFCS '13: 38th Symposium on Mathematical Foundations of Computer Science*, 813–824.

Leslie G. Valiant (1977). Graph-theoretic arguments in low-level complexity. In *MFCS '77: 6th Symposium on Mathematical Foundations of Computer Science*, volume 53 of *Lecture Notes in Computer Science*, 162–176. Springer.

Wolmer Vasconcelos (2005). *Integral closure.* Springer Monographs in Mathematics. Springer-Verlag, Berlin. Rees algebras, multiplicities, algorithms.

Virginia Vassilevska Williams (2012). Multiplying matrices faster than Coppersmith–Winograd. In *STOC '12: 44th Annual ACM Symposium on Theory of Computing*, 887–898. ACM.

Ryan Williams (2006). Inductive time-space lower bounds for SAT and related problems. *Comput. Complexity* **15**(4), 433–470.

Ryan Williams (2008). Time-space tradeoffs for counting NP solutions modulo integers. *Comput. Complexity* **17**(2), 179–219.

Ryan Williams (2013). Natural proofs versus derandomization. In *STOC '13: 45th Annual ACM Symposium on Theory of Computing*, 21–30. ACM, New York.

Ryan Williams (2014). Nonuniform ACC circuit lower bounds. *J. Assoc. Comput. Mach.* **61**(1), Art. 2, 32.

Andrew Chi-Chih Yao (1982). Theory and applications of trapdoor functions. In *FOCS '82: 23rd Annual IEEE Symposium on Foundations of Computer Science*, 80–91. IEEE, New York.

Andrew Chi-Chih Yao (1991). Lower bounds for algebraic computation trees with integer inputs. *SIAM J. Comput* **20**(4), 655–668.

Andrew Chi-Chih Yao (1997). Decision tree complexity and Betti numbers. *J. Comput. System Sci.* **55**(1, part 1), 36–43.

Norman Zadeh (1973). A bad network problem for the simplex method and other minimum cost flow algorithms. *Math. Programming* **5**, 255–266.

Manuscript received 16 June 2014

Joshua A. Grochow
Santa Fe Institute,
1399 Hyde Park Rd., Santa Fe,
NM 87501, USA.
jgrochow@santafe.edu