DETERMINISTIC POLYNOMIAL IDENTITY TESTS FOR MULTILINEAR BOUNDED-READ FORMULAE

Matthew Anderson, Dieter van Melkebeek,
and Ilya Volkovich

June 20, 2014

Abstract. We present a polynomial-time deterministic algorithm for testing whether constant-read multilinear arithmetic formulae are identically zero. In such a formula each variable occurs only a constant number of times and each subformula computes a multilinear polynomial. Our algorithm runs in time $s^{O(1)} \cdot n^{k^{O(k)}}$, where s denotes the size of the formula, n denotes the number of variables, and k bounds the number of occurrences of each variable. Before our work no subexponential-time deterministic algorithm was known for this class of formulae. We also present a deterministic algorithm that works in a blackbox fashion and runs in time $n^{k^{O(k)}+O(k \log n)}$ in general, and time $n^{k^{O(k^2)}+O(kD)}$ for depth D. Finally, we extend our results and allow the inputs to be replaced with sparse polynomials. Our results encompass recent deterministic identity tests for sums of a constant number of read-once formulae, and for multilinear depth-four formulae.

Keywords. Derandomization, Identity Testing, Arithmetic Circuits, Bounded-Depth Circuits

Subject classification. 12Y05, 68Q25

1. Introduction

Polynomial identity testing (PIT) denotes the fundamental problem of deciding whether a given polynomial identity holds. More precisely, we are given an arithmetic circuit or formula F on n inputs over a given field $\mathbb{F}$, and we wish to know whether all the coefficients of the formal polynomial P, computed by F, vanish. Due to its basic nature, PIT shows up in many constructions in the theory of computing. Particular problems that reduce to PIT include integer primality testing (Agrawal & Biswas 2003) and finding perfect matchings in graphs (Lovász 1979).

PIT has a very natural randomized algorithm – pick values for the variables uniformly at random from a small set S, and accept iff P evaluates to zero on that input. If $P \equiv 0$ then the algorithm never errs; if $P \not\equiv 0$ then by Schwartz-Zippel (DeMillo & Lipton 1978; Schwartz 1980; Zippel 1979) the probability of an error is at most $d/|S|$, where d denotes the total degree of P. This results in an efficient randomized algorithm for PIT. The algorithm works in a *blackbox* fashion in the sense that it does not access the representation of the polynomial P, rather it only examines the value of P at certain points (from $\mathbb{F}$ or an extension field of $\mathbb{F}$).

Despite the simplicity of the above randomized algorithm and much work over the course of thirty years, no efficient deterministic algorithm is known when the polynomial is given as an arithmetic formula.

Is there an efficient deterministic identity test for arithmetic formulae?

The question is central to the pursuit of circuit lower bounds. Efficiently derandomizing identity testing implies Boolean or arithmetic formula/circuit lower bounds that have been elusive for half a century (Aaronson & van Melkebeek 2011; Kabanets & Impagliazzo 2004; Kinne *et al.* 2012). In fact, an efficient deterministic blackbox identity test for arithmetic formulae of depth three or four already implies such lower bounds (Agrawal & Vinay 2008; Gupta *et al.* 2013). Conversely, the well-known hardness vs randomness tradeoffs imply that sufficiently strong Boolean circuit lower bounds yield efficient deterministic polynomial identity tests, and there are a couple of similar results starting from arithmetic circuit lower bounds as well (Dvir *et al.* 2009; Kabanets & Impagliazzo 2004).

The powerful connections with circuit lower bounds have energized much of the recent effort towards derandomizing identity testing for restricted classes of arithmetic formulae, in particular for constant-depth formulae. For depth-two formulae several deterministic polynomial-time blackbox algorithms are known (Agrawal 2003; Arvind & Mukhopadhyay 2010; Ben-Or & Tiwari 1988; Bläser *et al.* 2009; Klivans & Spielman 2001). For depth three the state of the art is a deterministic polynomial-time blackbox algorithm when the fanin of the top gate is fixed to any constant (Saxena & Seshadhri 2012). The same is known for depth four when the formulae are multilinear, i.e., when every gate in the formula computes a polynomial of degree at most one in each variable (Saraf & Volkovich 2011). There are also a few incomparable results for rather specialized classes of depth-four formulae (Arvind & Mukhopadhyay 2010; Saxena 2008; Shpilka & Volkovich 2009). We refer to the excellent survey papers (Saxena 2009; Shpilka & Yehudayoff 2010) for more information.

Another natural restriction is to bound the number of times each variable can occur in the formula. We call such a restricted formula *read-k*, where k denotes the maximum number of times each variable may appear. Identity testing for read-once formulae is trivial in the non-blackbox setting as there can be no cancellation of monomials. Shpilka and Volkovich considered a special type of bounded-read formulae, namely formulae that are the sum of k read-once formulae. For such formulae and constant k they established a deterministic polynomial-time non-blackbox identity test as well as a deterministic blackbox algorithm that runs in quasi-polynomial time, more precisely in time $s^{O(\log s)}$, on formulae of size s (Shpilka & Volkovich 2008, 2009). These results have been extended to sums of k read-once algebraic branching programs (Jansen *et al.* 2009). Algebraic branching programs are a model of computation analogous to Boolean branching programs and lying between formulae and circuits in terms of power.

1.1. Results. We present a deterministic polynomial-time identity test for multilinear constant-read formulae, as well as a deterministic quasi-polynomial-time blackbox algorithm for these formulae.

Theorem 1.1 (Main Result). *For each constant $k \in \mathbb{N}$ there is a deterministic polynomial identity test for multilinear read-k formulae of size s that runs in time* $\text{poly}(s)$*. In addition, there is a deterministic blackbox test that runs in time $s^{O(\log s)}$.*

Note that Theorem 1.1 extends the class of formulae that Shpilka and Volkovich could handle since a sum of read-once formulae is always multilinear. This is a *strict* extension; in Section 2.1.3 we exhibit an explicit multilinear read-twice formula with n variables that requires $\Omega(n)$ terms when written as a sum of read-once formulae. The separating example also shows that the efficiency of the identity test in Theorem 1.1 cannot be obtained by first expressing the given formula as a sum of read-once formulae and then applying the known algorithms for sums of read-once formulae (Shpilka & Volkovich 2009) to it.

Shpilka and Volkovich actually proved their result for sums of a somewhat more general type than read-once formulae, namely read-once formulae in which each leaf variable is replaced by a low-degree univariate polynomial in that variable. We can handle a further extension in which the leaf variables are replaced by *sparse multivariate* polynomials. We use the term *sparse-substituted* formula for a formula along with substitutions for the leaf variables by multivariate polynomials, where we assume that the substituted polynomials are each given as a list of terms (monomials). We call a sparse-substituted formula *read-k* if each variable appears in at most k of those multivariate polynomials. The substituted polynomials need not be multilinear, as long as for every multiplication gate of the original formula the different input branches of the gate are variable disjoint. We call such sparse-substituted formulae *structurally-multilinear.*

Theorem 1.2 (Extended Main Result). *For each constant $k \in \mathbb{N}$ there is a deterministic polynomial identity test for structurally-multilinear sparse-substituted read-k formulae that runs in time*

$s^{O(\log t)}$, where s denotes the size of the formula, and t the maximum number of terms a substitution consists of. In addition, there is a deterministic blackbox test that runs in time $s^{O(\log st)}$.

We observe that any multilinear depth-four alternating formula with an addition gate of fanin k as the output can be written as the sum of k sparse-substituted read-once formulae, where the read-once formulae are single monomials and the substitutions correspond to multilinear depth-two formulae. This implies that our blackbox algorithm also extends the work by Karnin *et al.* (2013), who established a deterministic quasi-polynomial-time blackbox algorithm for multilinear formulae of depth four. Thus, our results can be seen as unifying identity tests for sums of read-once formulae (Shpilka & Volkovich 2009) with identity tests for depth-four multilinear formulae (Karnin *et al.* 2013) while achieving comparable running times in each of those restricted settings. It remains an open question whether our results can be extended to test multilinear read-k algebraic branching programs in a way analogous to the extension from Shpilka & Volkovich (2009) to Jansen *et al.* (2009).

We can improve the running time of our blackbox algorithm in the case where the formulae have small depth. In particular, we obtain a polynomial-time blackbox algorithm for multilinear constant-read constant-depth formulae and, for the special case of fields with infinite characteristic, Agrawal *et al.* (2012) recently showed that the multilinearity condition can be removed. We refer to Section 6.3 for more details about those results, and to the rest of Section 6 for more general versions and finer parameterizations of our main result and its extensions.

1.2. Techniques. We now give an overview of our approach with a focus on the deterministic polynomial identity tests for multilinear read-k formulae given in Theorem 1.1. Both the blackbox and the non-blackbox algorithms are obtained by induction on k, using known algorithms in the base case of read-once formulae. To lift the identity tests for read-once formulae we exhibit reductions from testing multilinear read-$(k+1)$ formulae to testing multilinear read-k formulae. Applying the transformation k times thus reduces identity testing multilinear read-$(k+1)$ formulae to identity testing read-once formulae. At each stage the transformation consists of the following two steps, where the intermediate instances are sums of two multilinear read-k formulae, which we refer to as multilinear Σ^2-read-k formulae.

Step 1: Reduce multilinear read-$(k+1)$ formulae PIT to multilinear Σ^2-read-k formulae PIT.

Step 2: Reduce multilinear Σ^2-read-k formulae PIT to multilinear read-k formulae PIT.

We first explain the blackbox reductions, then discuss the more efficient but non-blackbox variant, and finally sketch the extension given by Theorem 1.2.

1.2.1. Blackbox setting. A deterministic blackbox polynomial identity test for a class $\mathcal{F}$ of formulae is known to be equivalent to a so-called *hitting set generator* for $\mathcal{F}$ (see, e.g., Shpilka & Yehudayoff (2010, Lemma 4.1)). The latter is a uniform collection of polynomial maps $\mathcal{G}_n : \mathbb{F}^{\ell(n)} \to \mathbb{F}^n$, one for every positive integer n, such that $\mathcal{G}_n$ hits all formulae from $\mathcal{F}$ on n variables, i.e., for every non-zero formula F in $\mathcal{F}$ on n variables, $F(\mathcal{G}_n)$ is a non-zero polynomial. We only need one direction of the equivalence, which follows from the Schwartz-Zippel lemma: Given a hitting set generator $\mathcal{G}_n$ of total degree $d_{\mathcal{G}_n}$ for $\mathcal{F}$, we can deterministically test whether a formula F in $\mathcal{F}$ of total degree d_F on n variables is identically zero by picking an arbitrary subset S of size $d_F \cdot d_{\mathcal{G}_n} + 1$ from $\mathbb{F}$ (or an extension field of $\mathbb{F}$) and checking that $F(\mathcal{G}_n(x)) = 0$ for every $x \in S^{\ell(n)}$. For a multilinear formula F, the total degree d_F is at most n, and the running time of the resulting blackbox identity test is $n^{O(\ell(n))}$ as long as the generator is computable in that amount of time, and has total degree $d_{\mathcal{G}_n}$ that is polynomially bounded.

Base case and generalization. For read-once formulae, Shpilka & Volkovich (2009) defined a polynomial-time computable polynomial map $G_{n,w} : \mathbb{F}^{2w} \to \mathbb{F}^n$ of total degree n over a field $\mathbb{F}$, and showed that $G_{n,\lceil \log n \rceil + 1}$ hits read-once formulae on n variables. We refer to $G_{n,w}$ as the *SV-generator*. By the above connection, the

SV-generator yields a deterministic blackbox identity test for read-once formulae that runs in time $n^{O(\log n)}$.

Using various additional properties of the SV-generator, we argue by induction on k that, for some function $f : \mathbb{N} \to \mathbb{N}$, $G_{n,f(k)+k\lceil \log n \rceil}$ hits multilinear read-k formulae on n variables. By the same token, this yields a blackbox identity test for multilinear read-k formulae that runs in time $n^{O(\log n)}$ for every fixed k. We now explain the two inductive steps in this setting.

Step 1. We show that if $G_{n,w}$ hits multilinear Σ^2-read-k formulae on n variables, then $G_{n,w+\lceil \log n \rceil}$ hits multilinear read-$(k+1)$ formulae on n variables. In order to do so, we exploit the property of the SV-generator that if $G_{n,w}$ hits a non-zero first-order partial derivative of a formula, then $G_{n,w+1}$ hits the formula itself. We show that for every non-constant multilinear read-$(k+1)$ formula F, there exists a variable x such that the partial derivative $\partial_x F$ of F with respect to x is non-zero and can be written as the product of (i) subformulae of F each depending on at most half of the variables, and (ii) a multilinear Σ^2-read-k formula which is the partial derivative of a subformula of F. We call this process of breaking up a formula via a well-chosen partial derivative *fragmentation* (this generalizes a technique of Karnin *et al.* (2013)). Note that the factors of type (i) are multilinear read-$(k+1)$ formulae themselves. By recursively fragmenting those factors $\lceil \log n \rceil$ times, they become constant and are hit by $G_{n,w}$ for trivial reasons. The factors of type (ii) are hit by $G_{n,w}$ by assumption. Since a hitting set generator for a class of formulae also hits all products of formulae from that class (because polynomial rings over fields are integral domains), inductively applying the above property of the SV-generator shows that $G_{n,w+\lceil \log n \rceil}$ hits the original multilinear read-$(k+1)$ formula F.

Step 2. We show that, for some function $h : \mathbb{N} \to \mathbb{N}$, if $G_{n,w}$ hits multilinear read-k formulae on n variables, then $G_{n,w+h(k)}$ hits multilinear Σ^2-read-k formulae on n variables. In order to do so, we exploit another property of the SV-generator. By a *zero-substitution* of a polynomial Q we mean the polynomial obtained from Q by setting some (possibly all or none) of the variables to zero. The SV-generator $G_{n,w'}$ has the property that it hits every non-zero polynomial Q on n variables that satisfies the following condition: no zero-substitution of Q equals a monomial of more than w' variables. The intuition for applying this property is that, no matter what the non-zero formula F looks like, for a random assignment σ the shifted formula $F(x+\sigma)$ satisfies the condition, and is therefore hit by $G_{n,w'}$. Moreover, the following key lemma shows that in the case of a multilinear Σ^2-read-k formula F, the only way the condition can fail is if σ happens to be a zero of one of the non-zero partial derivatives of small order of a nontrivial subformula of F.

Lemma 1.3 (Simplified Key Lemma). *There is a function $h : \mathbb{N} \to \mathbb{N}$ such that for every multilinear Σ^2-read-k formula F with variables x, and variable assignment σ where none of the non-zero partial derivatives of order $h(k)$ of any nontrivial subformula of F vanish, the zero-substitutions of $F(x + \sigma)$ are not monomials of more than $h(k)$ variables.*

Observe that all nontrivial subformulae of a multilinear Σ^2-read-k formula F are multilinear and read-k, as are their partial derivatives, because multilinear read-k formulae are closed under derivatives. Now, we assumed that $G_{n,w}$ hits multilinear read-k formulae on n variables, and therefore also products of such formulae. Thus, a common non-zero of the non-zero partial derivatives of order $h(k)$ of the nontrivial subformulae of F appears in the image of $G_{n,w}$. By Lemma 1.3 and the above property of the SV-generator, this means that for some σ in the image of $G_{n,w}$, the SV-generator $G_{n,w'}$ with $w' = h(k)$ hits the shifted formula $F(x + \sigma)$. This implies that $F(G_{n,w'} + G_{n,w}) \not\equiv 0$, where $G_{n,w'} + G_{n,w}$ denotes the component-wise sum of the polynomial maps $G_{n,w'}$ and $G_{n,w}$ on disjoint sets of variables. Yet another property of the SV-generator is its additivity: $G_{n,w'} + G_{n,w} = G_{n,w'+w}$. We conclude that $G_{n,w'+w}$ with $w' = h(k)$ hits F.

By combining the two steps, we conclude that if $G_{n,w}$ hits multilinear read-k formulae on n variables, then $G_{n,w+h(k)+\lceil \log n \rceil}$ hits multilinear read-$(k + 1)$ formulae on n variables. When we apply this combination $k - 1$ times starting from the SV-generator for read-once formulae, and simplify using the additivity property of the SV-generator, we obtain that $G_{n,f(k)+k\lceil \log n \rceil}$ hits multilinear read-k formulae on n variables, where $f(k) = \sum_{i=0}^{k-1} h(i)$. This yields our hitting set generator for multilinear read-k formulae.

Proving the Key Lemma. Let us briefly sketch the proof of Lemma 1.3. Let F be a multilinear Σ^2-read-k formula and let σ be an assignment for F of the type specified in the statement of the lemma. Ignoring zero substitutions for simplicity, suppose $F(x+\sigma)$ is identical to a monomial $M_V \doteq \prod_{i \in V} x_i$. The identity $F(x+\sigma) \equiv M_V$ must hold with respect to partial derivatives ∂_P for any set $P \subseteq V$. Moreover, $(\partial_P F)(x+\sigma) \equiv \partial_P(F(x+\sigma)) \equiv \partial_P M_V = M_{V \setminus P}$. We argue that provided $|V|$ is at least a sufficiently large function $h(k)$, there exists a set $P \subseteq V$ witnessing that $(\partial_P F)(x+\sigma) - M_{V \setminus P}$ is not identically zero. Similar to the approach of Karnin *et al.* (2013), our witness for non-identity is a structural one: We show that $\partial_P F$ can be rewritten in such a way that its structure alone indicates that $(\partial_P F)(x+\sigma)$ is not a monomial. The selection of P and the rewriting of $\partial_P F$ are components of the most intricate technical transformation in our paper. We call it *shattering* as it is accomplished, in part, by repeatedly fragmenting the summands of F. (Recall that fragmenting is our process of breaking up a formula via a well-chosen partial derivative.) We refer to Section 3.4 for more intuition about this key part of our paper.

1.2.2. Non-Blackbox Setting. When we are given access to the input formula itself, we can improve the running time for our deterministic identity test for multilinear read-k formulae from quasi-polynomial to polynomial.

In the non-blackbox setting, the case of read-once formulae is trivial. Since every variable occurs at most once, there is no cancellation of monomials at addition gates, and the only way a monomial can vanish is if it gets multiplied by a zero polynomial at a multiplication gate. If we iterate over the subformulae in a bottom-up fashion, this allows us to determine for each subformula whether it is zero. This yields a simple polynomial-time identity test for read-once formulae, which forms the base case for our inductive construction. We now explain how the two inductive steps mentioned above work in the non-blackbox setting.

Step 1. This reduction follows from a generalization of the simple algorithm for read-once formulae, based on the following intuition: Given a multilinear read-$(k+1)$ formula F, if an addition gate has an input g that contains all $k+1$ occurrences of some variable x, and g depends on x, then g cannot be cancelled at this gate. This implies that g can be replaced by a fresh variable without changing whether the overall polynomial is zero. For $k \geq 1$ this allows us to transform the formula in a bottom-up fashion into one where each addition gate is a multilinear Σ^2-read-k formula without affecting zeroness. During the process, we just need to be able to determine for each transformed addition gate whether it is zero, and if not, what variables it depends on. Both of these tasks reduce to PIT for multilinear Σ^2-read-k formulae. This yields a polynomial-time reduction from PIT for multilinear read-$(k+1)$ formulae to PIT for multilinear Σ^2-read-k formulae. This constitutes an improvement over the corresponding blackbox reduction which adds a logarithmic term to the seed length of the generator, and therefore a quasi-polynomial factor to the running time of the identity test.

Step 2. The blackbox version of this step is already efficient as each application adds only a constant amount to the seed length of the generator and therefore only a polynomial factor to the running time of the identity test. Thus, it suffices to simulate the behavior of the blackbox reduction on a multilinear Σ^2-read-k formula F with n variables. To this end we explicitly compute a common non-zero σ of the at most $n^{h(k)}$ non-zero partial derivatives of the subformulae of F up to order $h(k)$, using the assumed non-blackbox identity test for multilinear read-k formulae combined with the standard search-to-decision reduction for PIT. We conclude by testing F on $G_{n,h(k)} + \sigma$ over a domain of size polynomial in n. By the analysis of the blackbox case and Schwartz-Zippel, this is an identity test for F.

1.2.3. Extensions. To extend our results to multilinear *sparse-substituted* formulae, only a few modifications are needed. The most substantial change occurs in the fragmentation process, where we additionally use zero-substitutions to break up the sparse inputs to the formulae.

Our arguments hinge on multilinearity because (i) partial derivatives do not increase multilinear formula size, and (ii) the factors of multilinear formulae are variable disjoint. The relaxation to *structurally-multilinear* sparse-substituted formulae essentially maintains both of these properties. The effort in this extension comes in showing that an analog of the Key Lemma holds; we achieve this by carefully transforming structurally-multilinear sparse-substituted formulae into multilinear sparse-substituted formulae and then applying the original Key Lemma.

1.3. Organization. In Section 2 we introduce our notation and formally define the classes of arithmetic formulae that we study. Section 2 also reviews some preliminaries in more detail, including the SV-generator and our structural witnesses for non-zeroness akin to Karnin *et al.* (2013). In Section 3 we develop the Fragmentation Lemma in a step-wise fashion – for read-once formulae, read-k formulae, and sparse-substituted formulae – and the Shattering Lemma that is based on it. We develop our blackbox and non-blackbox identity tests in parallel. In Section 4 we reduce PIT for structurally-multilinear sparse-substituted read-$(k+1)$ formulae to PIT for structurally-multilinear sparse-substituted Σ^2-read-k formulae. In Section 5 we reduce PIT for structurally-multilinear sparse-substituted Σ^2-read-k formulae to PIT for structurally-multilinear sparse-substituted read-k formulae. Section 6 establishes identity tests for structurally-multilinear sparse-substituted read-k formulae and proves Theorems 1.1 and 1.2. We end with a specialization of our approach that gives a deterministic polynomial-time blackbox algorithm for multilinear constant-read constant-depth formulae.

2. Notation and Preliminaries

Let $\mathbb{F}$ denote a field, finite or otherwise, and let $\bar{\mathbb{F}}$ denote its algebraic closure. We assume that elements of $\mathbb{F}$ are represented in binary using some standard encoding. Moreover, we assume that there is an algorithm that given an integer r outputs in time $\mathrm{poly}(r)$ a set of r distinct elements in $\mathbb{F}$ (or an extension field of $\mathbb{F}$) each of which is represented in this encoding using $O(\log r)$ bits.

2.1. Polynomials and Arithmetic Formulae. A polynomial $P \in \mathbb{F}[x_1, ..., x_n]$ *depends* on a variable x_i if there are two inputs $\bar{\alpha}, \bar{\beta} \in \bar{\mathbb{F}}^n$ differing only in the i^{th} coordinate for which $P(\bar{\alpha}) \neq P(\bar{\beta})$. We denote by $\mathrm{var}(P)$ the set of variables that P depends on.

For a subset of the variables $X \subseteq \{x_1, ..., x_n\}$ and an assignment $\bar{\alpha}$, $P|_{X \leftarrow \bar{\alpha}}$ denotes the polynomial P with the variables in X substituted by the corresponding values in $\bar{\alpha}$. We often denote variables interchangeably by their index or by their label: i versus x_i, and $[n] \doteq \{1, 2, \ldots, n\}$ versus $\{x_1, ..., x_n\}$; we often drop the index and refer to $x \in X$.

An *arithmetic formula* is a tree where the leaves are labeled with variables or field elements and internal nodes (or gates) labeled with addition or multiplication. The singular gate with no outgoing wires is the output gate of the formula. We interchangeably use the notions of a gate and the polynomial computed by that gate.

The *size* of an arithmetic formula is the number of wires in the formula plus the total number of bits required to represent the constants. We assume that the encoding of the constants is such that size-s formulae containing no variables can be evaluated in time $\mathrm{poly}(s)$. The *depth* of an arithmetic formula is the length of a longest path from the output gate to an input variable. Except for the constant depth case we assume that the fanin of multiplication and addition gates is two.

2.1.1. Restricted Types of Arithmetic Formulae. An arithmetic formula is *multilinear* if every gate of the formula computes a polynomial that has degree at most one in every variable. This means that only one child of a multiplication gate may depend on a particular variable. However, more than one child may contain occurrences of some variable. For example, the formula

$$(x_1 - x_2) \cdot ((x_1 + x_3) - x_1)$$

is multilinear, and although the second factor has occurrences of x_1 it does not depend on x_1.

We also consider the restriction that each variable occurs only a bounded number of times.

Definition 2.1 (Read-k Formula). *For $k \in \mathbb{N}$, a* read-k *formula is an arithmetic formula that has at most k occurrences of each variable. For a subset $V \subseteq [n]$, a* read$_V$-k *formula is an arithmetic formula that has at most k occurrences of each variable in V (and an unrestricted number of occurrences of variables outside of V).*

Observe that for $V = [n]$ the notion of read$_V$-k coincides with read-k. One can build more complex formulae by adding several formulae together.

Definition 2.2 (Σ^m-Read-k Formula). *For $k, m \in \mathbb{N}$, a* Σ^m-read-k *formula is the sum of m read-k formulae.*

Note that any Σ^m-read-k formula is a read-(km) formula.

Bounded-read formulae can be generalized by replacing variables with sparse polynomials. We call a polynomial *t-sparse* if it consists of at most t terms.

Definition 2.3 (Sparse-Substituted Formula). *Let $B \in \mathbb{F}[y_1, \dots, y_r]$ be a read-once formula, and for $1 \le i \le r$ let $\rho_i \in \mathbb{F}[x_1, \dots, x_n]$ be a multivariate polynomial given as a list of terms. We call $F = B(\rho_1, \dots, \rho_r)$ a* sparse-substituted *formula with* backbone *B. A* subformula *of F is a formula of the form $f(\rho_1, \dots, \rho_r)$ where $f(y_1, \dots, y_r)$ is an input or internal gate of B, or a variable x_j occurring in F. Further,*

(i) for a subset $V \subseteq [n]$ if every variable $x_j \in V$ occurs in at most k of the ρ_i's, we say that F is read$_V$-k*,and*

(ii) if for every multiplication gate g in F and every variable x_j there is at most one multiplicand of g that depends on x_j, we say that F is structurally multilinear.

Note that for the notion of a subformula of a sparse-substituted formula F, we consider the sparse substitutions as atomic on the inputs x_j; we do not consider the individual terms of the sparse substitutions as subformulae of F. The individual terms do matter for the size of the sparse-substituted formula F, which is the size of the backbone formula B plus the size required to represent each sparse polynomial as a sum of at most t terms.

Note that a sparse-substituted formula F is multilinear if every gate, including the substituted input gates, computes a multilinear polynomial. This is equivalent to all multiplication gates in F having variable-disjoint children, and the sparse substitutions being multilinear. The corresponding interpretation of structural multilinearity is that the multiplication gates in F have variable-disjoint children, but the substituted sparse polynomials may not be multilinear. Thus, structural multilinearity is more general than multilinearity. For brevity we often drop the quantifier "sparse-substituted" when discussing structurally-multilinear formulae.

2.1.2. Partial Derivatives. Partial derivatives of multilinear polynomials can be defined formally over any field $\mathbb{F}$ by stipulating the partial derivative of monomials consistent with standard calculus, and imposing linearity. The well-known sum, product, and chain rules then carry over. For a multilinear polynomial $P \in \mathbb{F}[x_1, ..., x_n]$ and a variable x_i, we can write P uniquely as $P = Q \cdot x_i + R$, where $Q, R \in \mathbb{F}[x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n]$. In this case the partial derivative of P with respect to x_i is $\frac{\partial P}{\partial x_i} = Q$. We often shorten this notation to $\partial_{x_i} P$. Observe that $R = P|_{x_i \leftarrow 0}$.

For a multilinear read-k formula F, $\partial_x F$ is easily obtained from F, and results in a formula with a structure no more complex than F. Start from the output gate and recurse through the formula, applying at each gate the sum or product rule as appropriate. In the case of an addition gate $g = \sum_i g_i$, we have that $\partial_x g = \sum_i \partial_x g_i$. Thus, we recursively replace each of the children by their partial derivative. The structure of the formula is maintained, except that some children may disappear because they do not depend on x. In the case of a multiplication gate $g = \prod_i g_i$, we have $\partial_x g = \sum_i \partial_x g_i \cdot \prod_{j \neq i} g_j$. However, by the multilinearity condition at most one of the terms in the sum is non-zero because at most one g_i can depend on x. Thus, we leave the branches g_j for $j \neq i$ untouched, recursively replace g_i by its partial derivative. The structure of the formula is again maintained or simplified. Overall, the resulting formula $\partial_x F$ is multilinear and read-k. See Figure 2.1 for an example. Similarly, the partial derivatives of multilinear Σ^m-read-k formulae are multilinear Σ^m-read-k formulae. This leads to the following proposition.

Proposition 2.4. *Let F be a multilinear read-k formula, let x be a variable, and f a gate of F containing all occurrences of x in F. Then $\partial_x F$ is a multilinear read-k formula that can be written as $\partial_x F = \partial_x f \cdot \prod_{g \in U_F(f)} g$, where $U_F(f)$ denotes the unvisited children of the multiplication gates along the path from the output of F to f, i.e., those children that are not on the path themselves.*

Proof. That $\partial_x F$ is a multilinear read-k formula follows from the preceding discussion. To show that $\partial_x F$ can be written in the stated form, we proceed by induction on the length of the path from the output of F to f. In the base case, $f = F$ and the claim trivially holds. We now argue the induction step.

Suppose that $F = \sum_i F_i$ and, without loss of generality, that f is a descendant of F_1. By the sum rule and the fact that $x \notin \mathrm{var}(F_i)$ for $i \geq 2$ we have that $\partial_x F = \partial_x F_1$. The claim follows by the induction hypothesis since

$$\partial_x F_1 = \partial_x f \cdot \prod_{g \in U_{F_1}(f)} g$$

and the fact that $U_{F_1}(f) = U_F(f)$.

Now, suppose that $F = \prod_i F_i$ and, without loss of generality, that f is a descendant of F_1. By the product rule and the fact that $x \notin \mathrm{var}(F_i)$ for $i \geq 2$ we have that $\partial_x F = \partial_x F_1 \cdot \prod_{i \geq 2} F_i$. By the induction hypothesis $\partial_x F_1 = \partial_x f \cdot \prod_{g \in U_{F_1}(f)} g$. Noting that $U_F(f) = U_{F_1}(f) \cup \{F_i\}_{i \geq 2}$ completes the claim. □

To handle the case of structurally-multilinear formulae we extend the notion of partial derivative: $\partial_{x,\alpha} F \doteq F|_{x \leftarrow \alpha} - F|_{x \leftarrow 0}$ for some $\alpha \in \mathbb{F}$. Provided the size of $\mathbb{F}$ is more than the degree of x in the formula F, there exists some $\alpha \in \mathbb{F}$ such that $\partial_{x,\alpha} F \not\equiv 0$ iff F depends on x. For this more general definition the analogs of the sum and product rules follow for structurally-multilinear formulae. Given a structurally-multilinear formula F, $\partial_{x,\alpha} F$ can be computed by a structurally-multilinear formula with no larger size or read.

Partial derivatives have seen many applications in the study of arithmetic circuits. For example, they have been used to exhibit lower bounds (Baur & Strassen 1983; Mignon & Ressayre 2004;

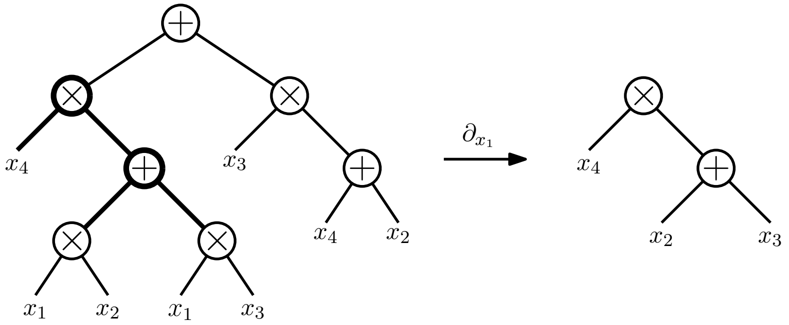

Figure 2.1: An example of taking the partial derivative of a multilinear read-twice formula.

Nisan & Wigderson 1996; Shpilka & Wigderson 2001), learn arithmetic circuits (Klivans & Shpilka 2006), and produce polynomial identity tests (Karnin *et al.* 2013; Shpilka & Volkovich 2008, 2009). In our setting, partial derivatives give us a handle on the structure of constant-read formulae, which we in turn exploit to develop our identity tests.

2.1.3. Linear Separation of Read-Twice and Σ-Read-Once Formulae. In this section we use partial derivatives to show a linear separation between read-twice and Σ-read-once formulae. We exhibit an explicit multilinear read-twice formula with n variables that requires $\Omega(n)$ terms when written as a sum of read-once formulae. In order to do so we follow an approach similar to that which Shpilka & Volkovich (2008, 2009) use to show "hardness of representation" results for sums of read-once formulae.

Consider some multilinear read-twice polynomial H_k which is purportedly computable by the sum of less than k read-once formulae, i.e., $H_k \equiv \sum_{i=1}^{k-1} F_i$. We argue that for an appropriate choice of H_k, some combination of partial derivatives and substitutions is sufficient to zero at least one of the branches F_i while not degrading the hardness of H_k by too much. Since H stays hard we can complete the argument by induction. In the base case H_1 is non-zero, so it requires at least one read-once formula to compute. This intuition is formalized in the following lemma.

Lemma 2.5. *For any non-trivial field $\mathbb{F}$ and each $k \in \mathbb{N}$ define*

$$H_k \doteq \prod_{i=1}^{2k-1} (x_{1,i}x_{2,i}x_{3,i} + x_{1,i} + x_{2,i} + x_{3,i}).$$

H_k is a multilinear read-twice formula which depends on $6k-3$ variables and is not computable by the sum of less than k read-once formulae.

Proof. Observe that for all $i \in \mathbb{N}$, $(x_{1,i}x_{2,i}x_{3,i} + x_{1,i} + x_{2,i} + x_{3,i})$ is a multilinear read-twice formula. Therefore for all $k \in \mathbb{N}$, H_k is a multilinear read-twice formula. We prove the second half of the claim by induction. When $k = 1$, H_1 is non-zero and hence the claim holds trivially. Now consider the induction step. Suppose the contrary: There exists a sequence of at most $k-1$ read-once formulae $\{F_i\}$ such that $H_k = \sum_{i=1}^{k-1} F_i$.

Consider F_{k-1}. Suppose there exists a pair of variables y, z such that $\partial_{y,z} F_{k-1} \equiv 0$. These operations modify at most two factors of H_k but do not zero them. Therefore $\partial_{y,z} H_k = H' \cdot H_{k-1}$ for some non-zero multilinear read-twice formula H' that depends on four variables and is variable disjoint from H_{k-1} (abusing notation to relabel the variables). Since $H' \not\equiv 0$ and multilinear, there exists $\bar{\beta} \in \{0,1\}^4 \subseteq \mathbb{F}^4$ such that $H'(\bar{\beta}) = c \neq 0$. This means that $\partial_{y,z} H_k|_{\text{var}(H') \leftarrow \bar{\beta}} = c \cdot H_{k-1}$. Hence H_{k-1} can be written as $\sum_{i=1}^{k-2} c^{-1} \partial_{y,z} F_i|_{\text{var}(H') \leftarrow \bar{\beta}}$, which contradicts the induction hypothesis. Therefore we can assume that for all pairs of variables y and z, $\partial_{y,z} F_{k-1} \not\equiv 0$.

This together with the read-once property of F_{k-1} implies the that least common ancestor of any pair of variables in F_{k-1} must exist and must be a multiplication gate. This also implies that F_{k-1} depends on all variables in H_k. Consider some variable y. Now, since $k > 1$ there must exist a variable z such that the least common ancestor of y and z in F_{k-1} is the first multiplication gate above y which depends on a variable other than y. Because F_{k-1} is a read-once formula we can write $\partial_z F_{k-1} = (y - a) \cdot F'_{k-1}$ for some $a \in \mathbb{F}$ and a read-once formula F'_{k-1} which is independent of y and z. Therefore $(\partial_z F_{k-1})|_{y \leftarrow a} \equiv 0$. By inspection we see that for all variables y, z and $a \in \mathbb{F}$, $(\partial_z H_k)|_{y \leftarrow a} = H' \cdot H_{k-1}$ for some non-zero multilinear read-twice formula H' which is variable disjoint from H_{k-1}. By the argument in the previous paragraph we may again conclude by contradicting the induction hypothesis. □

This implies the following corollary.

Corollary 2.6. *There exists a multilinear read-twice formula in n variables such that all k sums of read-once formula computing it require $k = \Omega(n)$.*

2.1.4. From Structurally-Multilinear to Multilinear Sparse-Substituted. In this subsection we exhibit an efficiently-computable transformation $\mathcal{L}$ from folklore that takes a structurally-multilinear formula and produces a multilinear sparse-substituted formula while preserving non-zeroness. We will use it in Section 4, Section 5, and Section 6 to extend our results for multilinear sparse-substituted formulae to structurally-multilinear sparse-substituted formulae.

For set of variables $X \doteq \{x_1, \ldots, x_n\}$ we define $\mathfrak{X} \doteq \left\{x_j^d \mid j, d \geq 1\right\}$ to be the set of all positive powers of the variables in X. Consider a new set of variables $Y \doteq \{y_{j,d} \mid j, d \geq 1\}$, and observe that there is a bijection between $\mathfrak{X}$ and Y. The transformation $\mathcal{L}$ maps elements of $\mathfrak{X}$ into variables of Y in a natural way.

Definition 2.7 (The transformation $\mathcal{L}$). *Let $X \doteq \{x_1, \ldots, x_n\}$ and $Y \doteq \{y_{j,d} \mid j, d \geq 1\}$. Let $f \in \mathbb{F}[X]$ be a sparse-substituted formula.*

- *For $j, d \geq 1$, let $\mathcal{L}_{\left\{x_j^d\right\}}(f)$ be the result of replacing every occurrence of exactly x_j^d in each term of a sparse-substituted input of f by the variable $y_{j,d}$.*
- *Let A be a set of positive powers of variables in X. Let $\mathcal{L}_A(f)$ be the result of applying $\mathcal{L}_{\left\{x_j^d\right\}}$ to f for all $x_j^d \in A$. Furthermore, let $\mathcal{L}(f)$ denote the result of taking A to be the set of all positive powers of every variable in X, e.g., the result of replacing all positive powers of x_j-variables by the corresponding $y_{j,d}$'s.*

- *For any set $P \subseteq Y$, let $X(P) \doteq \left\{x_j^d \mid y_{j,d} \in P\right\}$ be the preimages of the $y_{j,d}$'s under $\mathcal{L}$.*

For concreteness we give a few examples of the transformation $\mathcal{L}$ being applied to structurally-multilinear formulae:

$$\mathcal{L}(x_1^2 x_3) = y_{1,2} y_{3,1},$$
$$\mathcal{L}\left((x_1^2 x_3 + x_1 x_3^6) \cdot (x_2^3 x_4 + 3)\right) = (y_{1,2} y_{3,1} + y_{1,1} y_{3,6}) \cdot (y_{2,3} y_{4,1} + 3).$$

The following lemma demonstrates the connection between a formula f and its transformation $\mathcal{L}(f)$. The lemma exploits the fact that in a structurally-multilinear formula variables are never multiplied with themselves outside a sparse-substituted input. This implies that we can treat each degree of x_j as if it were a distinct variable. Additionally, we observe that setting $x_j \leftarrow a$ in f, for some $a \in \mathbb{F}$, is equivalent to setting $\left\{y_{j,d} \leftarrow a^d \mid d \geq 1\right\}$ in $\mathcal{L}(f)$.

Lemma 2.8. *Let $f \in \mathbb{F}[X]$ be a structurally-multilinear sparse-substituted read-k formula. Let $P, Z \subseteq Y$ be two disjoint subsets of variables and let $\bar{\sigma} \in \bar{\mathbb{F}}^n$ be an assignment. Then the following holds:*

(i) $\mathcal{L}(f)$ is a multilinear sparse-substituted read-k formula.

(ii) $f \equiv 0$ if and only if $\mathcal{L}(f) \equiv 0$.

(iii) $\partial_P(\mathcal{L}_{X(P\cup Z)}(f))|_{Z\leftarrow 0}$ does not depend on any $y_{j,d}$.

(iv) $(\partial_P(\mathcal{L}(f))|_{Z\leftarrow 0})\,|_{\left\{y_{j,d}\leftarrow \sigma_j^d \,\mid\, j,d\geq 1\right\}}$
$= \left(\partial_P(\mathcal{L}_{X(P\cup Z)}(f))|_{Z\leftarrow 0}\right)|_{\{x_j\leftarrow\sigma_j \,\mid\, j\geq 1\}}$

Proof. We first demonstrate a useful property of $\mathcal{L}$, and then show that it implies the properties stated in the lemma. Consider a term $T = c \cdot \prod_{j=1}^n x_j^{d_j}$ in the expansion of f. Each such term is produced by the sum of various products of terms from the sparse-substituted inputs:

$$T \equiv \sum_i c_i \prod_{j=1}^n x_j^{d_j} = \sum_i \prod_r T_{ir},$$

where each T_{ir} is a term from a sparse-substituted input. We can assume that for each i, the terms T_{ir} are all from different sparse-substituted inputs. Since f is structurally multilinear, for each i, the terms T_{ir} are variable disjoint, and hence each variable may occur in at most one factor T_{ir}.

Consider $\mathcal{L}_{x_j^d}(T)$. If $x_j^d \mid T$, but $x_j^{d+1} \nmid T$, then $\mathcal{L}_{x_j^d}(T) = y_{j,d} \cdot T/x_j^d$. Otherwise $\mathcal{L}_{x_j^d}(T) = T$. This is a 1-1 mapping on terms, and linearly extends to the sum of terms forming the expansion of a structurally-multilinear sparse-substituted formula f. Moreover, for any set of variable powers A, $\mathcal{L}_A$ maps the terms of a structurally-multilinear sparse-substituted formula in a 1-1 way.

We now prove the properties claimed by the lemma.

Part 1. $\mathcal{L}(f)$ is multilinear, because for each term and variable power in the expansion of f, the exact variable power x_j^d is replaced by a $y_{j,d}$. $\mathcal{L}(f)$ is a multilinear sparse-substituted formula because the transformation is performed on each sparse-substituted input individually. $\mathcal{L}(f)$ is read-k because each $y_{j,d}$ occurs in no more sparse-substituted inputs of $\mathcal{L}(f)$ than x_j does in f.

Part 2. We demonstrated that $\mathcal{L}$ induces a 1-1 correspondence between the terms of f and $\mathcal{L}(f)$. Moreover, non-zero terms are mapped to non-zero terms. Hence $f \equiv 0$ iff $\mathcal{L}(f) \equiv 0$.

Part 3. By definition the y variables in $\mathcal{L}_{X(P\cup Z)}(f)$ are in $P \cup Z$. The conclusion follows because partial derivatives and substitutions eliminate all dependence on the variables they act on.

Part 4. This property follows from two claims, which hold for any structurally-multilinear sparse-substituted formula g:

(i) $\mathcal{L}(g)|_{\left\{y_{j,d} \leftarrow \sigma_j^d \,\mid\, j,d \geq 1\right\}} = g|_{\{x_j \leftarrow \sigma_j \,\mid\, j \geq 1\}}$, and

(ii) $\partial_P \mathcal{L}(g)|_{Z\leftarrow 0} \equiv \mathcal{L}(\partial_P \mathcal{L}_{X(P\cup Z)}(g)|_{Z\leftarrow 0})$.

Claim (i) follows immediately from the 1-1 mapping between terms of g and $\mathcal{L}(g)$ established above. To see claim (ii) we argue that for all constants c:

$$\mathcal{L}(g)|_{y_{j,d}\leftarrow c} \equiv \mathcal{L}(\mathcal{L}_{x_j^d}(g)|_{y_{j,d}\leftarrow c}). \tag{2.9}$$

This essentially says that substitutions for y variables can be moved ahead of most of the transformation done by $\mathcal{L}$. Consider a term T in the expansion of g. If $x_j^d \mid T$, but $x_j^{d+1} \nmid T$, then $\mathcal{L}_{x_j^d}(T) = y_{j,d} \cdot T/x_j^d$ and

$$\begin{aligned}\mathcal{L}(T)|_{y_{j,d}\leftarrow c} &\equiv (y_{j,d} \cdot \mathcal{L}(\frac{T}{x_j^d}))|_{y_{j,d}\leftarrow c} \equiv c \cdot \mathcal{L}(\frac{T}{x_j^d}) \equiv \mathcal{L}(c \cdot \frac{T}{x_j^d}) \\ &\equiv \mathcal{L}((y_{j,d} \cdot \frac{T}{x_j^d})|_{y_{j,d}\leftarrow c}) \equiv \mathcal{L}(\mathcal{L}_{x_j^d}(T)|_{y_{j,d}\leftarrow c}).\end{aligned}$$

Otherwise, $\mathcal{L}(T)$ does not depend on $y_{j,d}$, then $\mathcal{L}(T|_{y_{j,d}\leftarrow c}) = \mathcal{L}(T)$, and therefore T contributes equally to both sides of Equation (2.9). By linearity we have Equation (2.9). Claim (ii) follows by performing similar analysis for partial derivatives. This completes the proof of Part 4 and the lemma. □

2.1.5. Polynomial Identity Testing and Hitting Set Generators. Arithmetic formula identity testing denotes the problem of deciding whether a given arithmetic formula is identically zero as a formal polynomial. More precisely, let F be an arithmetic formula on n variables over the field $\mathbb{F}$. The formula F is identically zero iff all coefficients of the formal polynomial that F defines vanish. For example, the formula $(x-1)(x+1)-(x^2-1)$ is identically zero but the formula $x^2 - x$ is non-zero (even over the field with two elements).

There are two general paradigms for identity testing algorithms: *blackbox* and *non-blackbox*. In the non-blackbox setting, the algorithm is given the description of the arithmetic formula as input. In the blackbox setting, the algorithm is allowed only to make queries to an oracle that evaluates the formula on a given input. Observe that non-blackbox identity testing reduces to blackbox identity testing because the description of a formula can be used to efficiently evaluate the formula on each query the blackbox algorithm makes. There is one caveat – in the blackbox case the algorithm should be allowed to query inputs from a sufficiently large field. This may be an extension field if the base field is too small. Otherwise, it is impossible to distinguish a polynomial that is functionally zero over $\mathbb{F}$ but not zero as a formal polynomial, from the formal zero polynomial (e.g., $x^2 - x$ over the field with two elements).

Blackbox algorithms for a class $\mathcal{P}$ of polynomials naturally produce a *hitting set*, i.e., a set H of points such that each non-zero polynomial $P \in \mathcal{P}$ from the class does not vanish at some point in H. In this case we say that H *hits* the class $\mathcal{P}$, and each P in particular. To see the connection, observe that when a blackbox algorithm queries a point that is non-zero it can immediately stop. Conversely, when the result of every query is zero, the algorithm must conclude that the polynomial is zero; otherwise, it fails to correctly decide the zero polynomial.

A related notion is that of a *hitting set generator*. Formally, a polynomial map $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n)$ where each $\mathcal{G}_i \in \mathbb{F}[y_1, y_2, \ldots, y_\ell]$ is a hitting set generator (or generator for short) for a class $\mathcal{P}$ of polynomials on n variables, if for each non-zero polynomial $P \in \mathcal{P}$, $\mathcal{G}$ *hits* P, that is the composition of P with $\mathcal{G}$ (denoted $P(\mathcal{G})$) is non-zero. Suppose that $\mathcal{G}$ hits a class of polynomials $\mathcal{P}$, then $\mathcal{G}$ can be used to construct a blackbox identity test for $P \in \mathcal{P}$ by collecting all elements in the image of $\mathcal{G}$ when we let the input variables to $\mathcal{G}$ range over some small set.

PROPOSITION 2.10. *Let $\mathcal{P}$ be a class of n-variate polynomials of total degree at most d. Let $\mathcal{G} \in (\mathbb{F}[y_1, ..., y_\ell])^n$ be a generator for $\mathcal{P}$ such that the total degree of each polynomial in $\mathcal{G}$ is at most $d_\mathcal{G}$ and $\mathcal{G}$ can be evaluated on elements of representation size q in time $T(q)$. There is a deterministic blackbox polynomial identity test for $\mathcal{P}$ that runs in time $(d \cdot d_\mathcal{G})^{O(\ell)} \cdot T(\log(O(d \cdot d_\mathcal{G})))$ and queries points from an extension field of size $O(d \cdot d_\mathcal{G})$.*

PROOF. Let P be a non-zero polynomial in $\mathcal{P}$. Since $\mathcal{G}$ is a generator for $\mathcal{P}$, the polynomial $P(\mathcal{G}) \in \mathbb{F}[y_1, y_2, \ldots, y_\ell]$ is non-zero. The total degree of $P(\mathcal{G})$ is at most $d \cdot d_\mathcal{G}$. By the Schwartz-Zippel Lemma (DeMillo & Lipton 1978; Schwartz 1980; Zippel 1979) any set $V^\ell \subseteq \mathbb{E}^\ell$, where $|V| \geq d \cdot d_\mathcal{G} + 1$, and $\mathbb{E}$ is an extension field of $\mathbb{F}$, contains a point at which $P(\mathcal{G})$ does not vanish. Note that the extension field $\mathbb{E} \supseteq \mathbb{F}$ must be sufficiently large to support the subset V of the required size. The algorithm tests P at all points in $\mathcal{G}(V^\ell)$ and outputs zero iff all test points are zero. □

Note that this approach is only efficient when $\ell \ll n$, the degrees are not too large, and $\mathcal{G}$ can be efficiently evaluated.

Hitting set generators and hitting sets are closely related. By Proposition 2.10 a hitting set generator implies a hitting set. It is also known that a hitting set generator can be efficiently constructed from a hitting set using polynomial interpolation (Shpilka & Volkovich 2009).

2.2. SV-Generator. One example of such a generator is the one Shpilka and Volkovich obtained by interpolating (i.e., passing a low-degree curve through) the set of all points in $\{0,1\}^n$ with at most w non-zero components. The resulting generator $G_{n,w}$ is a polynomial map of total degree n on $2w$ variables. Shpilka & Volkovich (2009) showed that it hits Σ^k-read-once formulae for $w \geq 3k + \log n$. Karnin *et al.* (2013) also used it to construct a hitting set generator for multilinear depth-four formulae with bounded top fanin.

For completeness, we include the definition of the generator $G_{n,w}$.

Definition 2.11 (SV-Generator Shpilka & Volkovich 2009). *Let $a_1, \ldots, a_n$ denote n distinct elements from a field $\mathbb{F}$, and for $i \in [n]$ let $L_i(x) \doteq \prod_{j \neq i} \frac{x-a_j}{a_i-a_j}$ denote the corresponding Lagrange interpolant. For every $w \in \mathbb{N}$, define $G_{n,w}(y_1, \ldots, y_w, z_1, \ldots, z_w)$ as*

$$\left(\sum_{j=1}^{w} L_1(y_j) z_j, \sum_{j=1}^{w} L_2(y_j) z_j, \ldots, \sum_{j=1}^{w} L_n(y_j) z_j \right).$$

Let $(G_{n,w})_i$ denote the i^{th} component of $G_{n,w}$; we refer to a_i as the Lagrange constant *associated with this i^{th} component.*

For intuition, when $w = 1$ we can view $G_{n,1}(y_1, z_1)$ as hashing z_1 into one of n buckets, where the bucket is determined by y_1. For general w, $G_{n,w}(y_1, \ldots, y_w, z_1, \ldots, z_w)$ can be regarded as hashing variables $z_1, \ldots, z_w$ into n buckets determined by $y_1, \ldots, y_w$. The value assigned to a bucket is the sum of the variables that are hashed into it. Note that this interpretation is only accurate when the values of the y_j's are all among the Lagrange constants $a_1, \ldots, a_n$.

For two polynomial maps $\mathcal{G}_1$ and G_2 with the same output length, it is natural to consider their component-wise sum. We denote this sum by $\mathcal{G}_1 + \mathcal{G}_2$, where we implicitly assume the variables of $\mathcal{G}_1$ and $\mathcal{G}_2$ have been relabelled so as to be *disjoint*. In probabilistic terms this corresponds to taking independent samples from $\mathcal{G}_1$ and $\mathcal{G}_2$, and adding them component-wise. With this convention in mind, the SV-generator has a number of useful properties that follow immediately from its definition.

PROPOSITION 2.12 (Karnin *et al.* 2013, Observations 4.2, 4.3). *Let w, w' be positive integers.*

(i) Every $\bar{\mu} \in \bar{\mathbb{F}}^n$ with at most w non-zero components is in the image of $G_{n,w}$.

*(ii) $G_{n,w}(y_1, \dots, y_w, z_1, \dots, z_w)|_{y_w \leftarrow a_i}$
$= G_{n,w-1}(y_1, \dots, y_{w-1}, z_1, \dots, z_{w-1}) + z_w \cdot \bar{e}_i$,
where $\bar{e}_i$ is the 0-1-vector with a single 1 in position i and a_i the i^{th} Lagrange constant.*

(iii)
$$\begin{array}{rr} & G_{n,w}(y_1, \dots, y_w, z_1, \dots, z_w) \\ + & G_{n,w'}(y_{w+1}, \dots, y_{w+w'}, z_{w+1}, \dots, z_{w+w'}) \\ \hline = & G_{n,w+w'}(y_1, \dots, y_w, \dots, y_{w+w'}, z_1, \dots, z_w, \dots, z_{w+w'}) \end{array}$$

The first item formalizes the property that the SV-generator interpolates the set of all points with at most w non-zero components. The second item shows how to make a single output component (and no others) depend on a particular z_j. The final item shows that sums of independent copies of the SV-generator are equivalent to a single copy of the SV-generator with the appropriate parameter w. Proposition 2.12 implies the following.

PROPOSITION 2.13. *Let $P = \sum_d P_d x_i^d$, where each $P_d \in \mathbb{F}[x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n]$ is a polynomial independent of the variable x_i. Suppose the polynomial map $\mathcal{G}$ hits P_d for some $d > 0$, then $P(\mathcal{G} + G_{n,1})$ is non-constant.*

PROOF. Consider $\mathcal{G} + G_{n,1}$. Without loss of generality, the variables in $G_{n,1}$ are y_1 and z_1, and by convention are disjoint from the variables in $\mathcal{G}$. Set the seed variable y_1 to the Lagrange constant a_i associated with x_i. By Proposition 2.12, Part (ii), $G_{n,1}(y_1, z_1)|_{y_1 \leftarrow a_i} = z_1 \cdot \bar{e}_i$.

Now, consider P composed with $\mathcal{G} + G_{n,1}(a_i, z_1)$ and write:

$$P(\mathcal{G} + G_{n,1})|_{y_1 \leftarrow a_i} = P(\mathcal{G} + G_{n,1}(a_i, z_1)) = \sum_d P_d(\mathcal{G})((\mathcal{G})_i + z_1)^d.$$

By hypothesis $P_d(\mathcal{G}) \not\equiv 0$ for some $d > 0$, fix j to be the maximum such index. Since $\mathcal{G}$ is independent of z_1, $P_j(\mathcal{G}) \not\equiv 0$, and j is maximal: $P(\mathcal{G} + G_{n,1}(a_i, z_1))$ has a monomial which depends on z_1^j that cannot be canceled. Therefore $P(\mathcal{G} + G_{n,1}(a_i, z_1))$ is non-constant and hence $P(\mathcal{G} + G_{n,1})$ is as well. □

This proposition implies the following connection between the SV-generator and partial derivatives.

Lemma 2.14. *Let P be a polynomial, x be a variable, and $\alpha \in \mathbb{F}$. If $\mathcal{G}$ hits a non-zero $\partial_{x,\alpha} P$, then $P(\mathcal{G} + G_{n,1})$ is non-constant.*

Proof. Write P as a univariate polynomial in x:

$$P = \sum_{i=0}^{d} P_i x^i,$$

where the polynomials P_i do not depend on x. By definition

$$\partial_{x,\alpha} P = P|_{x \leftarrow \alpha} - P|_{x \leftarrow 0} = \sum_{i=1}^{d} P_i \alpha^i.$$

Our hypothesis $(\partial_{x,\alpha} P)(\mathcal{G}) \not\equiv 0$ then implies that there is a $j > 0$ such that $P_j(\mathcal{G}) \not\equiv 0$. Applying Proposition 2.13 completes the proof. □

We now use this lemma to argue that the SV-generator hits sparse polynomials. Consider a sparse polynomial F. For any variable x that does not divide F, either at least half of the terms of the sparse polynomial depend on x, or at least half of the terms do not. In the former situation setting x to zero eliminates at least half of the terms in F; in the latter situation taking the partial derivative with respect to x has the same effect. Combining this with Lemma 2.14 and the properties of the SV-generator completes the argument.

LEMMA 2.15. *Let F be a non-constant sparse polynomial on n variables with t terms. $F(G_{n,\lceil \log t \rceil+1})$ is non-constant.*

PROOF. Assume, without loss of generality, that there are no duplicate monomials present in F. We proceed by induction on t. Suppose $t = 1$. By hypothesis F consists of a single non-constant monomial. Because the components of $G_{n,1}$ are non-constant we can conclude that $F(G_{n,1})$ is non-constant. Now consider the induction step for $t > 1$. Let $w \doteq \lceil \log \frac{t}{2} \rceil + 1$.

Case 1: Suppose there exists a variable $x_i \in \mathrm{var}(F)$ such that at most half of the terms depend on x_i. Then there is an $\alpha \in \bar{\mathbb{F}}$ such that $\partial_{x_i,\alpha} F \not\equiv 0$ and has at most $\frac{t}{2}$ terms. By induction $(\partial_{x_i,\alpha} F)(G_{n,w}) \not\equiv 0$. By Lemma 2.14, $F(G_{n,w} + G_{n,1})$ is non-constant. Applying Proposition 2.12, Part (iii), completes the case.

Case 2: Otherwise, for each variable $x_i \in \mathrm{var}(F)$ more than half the terms in F depend on x_i. There are two cases.

A. Suppose there exists a variable $x_i \in \mathrm{var}(F)$ such that $F|_{x_i \leftarrow 0}$ is non-constant. We argue that $F(G_{n,w+1}) = F(G_{n,w}(y_1, \ldots, y_w, z_1, \ldots, z_w) + G_{n,1}(y_{w+1}, z_{w+1}))$ is non-constant. To see this, consider setting y_{w+1} to the i^{th} Lagrange constant a_i and $z_{w+1} = -(G_{n,1})_i$. Because F is a sparse polynomial it may be written as $F = F_{x_i} \cdot x_i + F|_{x_i \leftarrow 0}$, for some sparse polynomial F_{x_i} which may depend on x_i. By Proposition 2.12, Part (ii):

$$\begin{aligned}
&F(G_{n,w+1})|_{y_{w+1} \leftarrow a_i,\; z_{w+1} \leftarrow -(G_{n,w})_i} \\
&\quad = F(G_{n,w} + G_{n,1}(a_i, -(G_{n,w})_i)) \\
&\quad = F|_{x_i \leftarrow 0}(G_{n,w}) \\
&\qquad + F_{x_i}(G_{n,w} + G_{n,1}(a_i, -(G_{n,w})_i)) \cdot \underbrace{((G_{n,w})_i - (G_{n,w})_i)}_{\equiv 0} \\
&\quad = F|_{x_i \leftarrow 0}(G_{n,w}).
\end{aligned}$$

By induction, the RHS of the above equation is non-constant, and hence $F(G_{n,w+1})$ is non-constant.

B. Otherwise, for all $x_i \in \mathrm{var}(F)$, $F|_{x_i \leftarrow 0}$ is a constant. We can assume without loss of generality that F is not divisible by any variable because such a variable can be factored out and independently hit by $G_{n,w+1}$. Therefore, for each $x_i \in \mathrm{var}(F)$ at least one term of F does not depend on x_i. Combining this fact with the hypothesis of the case implies, without loss of generality, that F has a non-zero constant term c. We can write $F = F' + c$ for a non-constant sparse polynomial F' with $t-1$ terms. By induction $F'(G_{n,w+1})$ is non-constant. Hence $F(G_{n,w+1})$ is non-constant.

This completes the proof. □

Before arguing the last necessary property of the SV-generator, we state one additional definition.

Definition 2.16. *For $\ell \in \mathbb{N}$, let $\mathcal{D}_\ell$ denote the class of non-zero polynomials that are divisible by a multilinear monomial on ℓ variables, i.e., the product of ℓ distinct variables. We use M_ℓ to denote the monomial $\prod_{i=1}^{\ell} x_i$.*

We require a property of the SV-generator that is implicit in Shpilka & Volkovich (2008, 2009). Informally it states: If a class of polynomials is disjoint from $\mathcal{D}_\ell$, and is closed under zero substitution, then the SV-generator hits this class of polynomials.

Lemma 2.17 (Implicit in Shpilka & Volkovich 2009, Theorem 6.2). *Let $\mathcal{P}$ be a class of polynomials on n variables that is closed under zero-substitutions. If $\mathcal{P}$ is disjoint from $\mathcal{D}_\ell$ for every $\ell > w$, the map $G_{n,w}$ is a hitting set generator for $\mathcal{P}$.*

Proof. Fix P in $\mathcal{P}$, and let d denote the maximum degree of individual variables in P. Let $S \subseteq \bar{\mathbb{F}}$ with $|S| = d+1$ and $0 \in S$. Define the set H^n_w to be the set of vectors in S^n with at most w non-zero components. By Proposition 2.12, Part (i) the set H^n_w is in the image of $G_{n,w}$.

Since the image of $G_{n,w}$ contains H^n_w, it is sufficient to prove that $P|_{H^n_w} \equiv 0$ implies $P \equiv 0$. For the given value of d, we prove the latter statement by induction on n. If $n \leq w$, then H^n_w is all of S^n. Since P has individual degree at most d, there is point in S^n which witnesses the non-zeroness of P. Therefore, $P|_{H_w^n} \equiv 0$ implies $P \equiv 0$, completing the base case.

Now, consider $n > w$ and suppose that $P|_{H_w^n} \equiv 0$. For some $i \in [n]$, let $P' \doteq P|_{x_i \leftarrow 0}$. By the closure under zero-substitutions of $\mathcal{P}$, $P' \in \mathcal{P}$. Since H_w^{n-1} is a projection of $H_w^n \cap \{\bar{\mu} \in S^n | \mu_i = 0\}$, we have that $P'|_{H_w^{n-1}} \equiv 0$. The individual degree of P' is at most d, and P' depends on at most $n-1$ variables. By the induction hypothesis $P|_{x_i \leftarrow 0} = P' \equiv 0$ and therefore $x_i|P$. The above argument works for any $i \in [n]$, so $x_i|P$ for all $i \in [n]$. Hence, $(\prod_{i=1}^n x_i)|P$. We have that $P = Q \cdot \prod_{i=1}^n x_i$ for some polynomial Q. Since $P \in \mathcal{P}$ and $\mathcal{P} \cap \mathcal{D}_n = \emptyset$ for $n > w$, we conclude that $Q \equiv 0$. Thus $P \equiv 0$, completing the proof. □

2.3. Structural Witnesses. Derandomizing polynomial identity testing means coming up with deterministic procedures that exhibit witnesses for non-zero circuits. The most obvious type of witness consists of a point where the polynomial assumes a non-zero value; such witnesses are used in blackbox tests. For restricted classes of circuits one may hope to exploit their structure and come up with other types of witnesses. The prior deterministic identity tests we mentioned (Karnin *et al.* 2013; Karnin & Shpilka 2008; Kayal & Saraf 2009; Saraf & Volkovich 2011; Saxena & Seshadhri 2009; Shpilka & Volkovich 2008, 2009) follow the latter general outline. More specifically, they exhibit a measure for the complexity of the restricted circuit that can be efficiently computed when given the circuit as input, and prove that (i) restricted circuits that are zero have low complexity, and (ii) restricted circuits of low complexity are easy to test. This framework immediately yields a non-blackbox identity test for the restricted class of circuits, and in several cases also forms the basis for a blackbox algorithm. Complexity measures that have been successfully used within this framework are the rank of depth-three circuits (Dvir & Shpilka 2007; Karnin *et al.* 2013; Karnin & Shpilka 2008) and the sparsity of multilinear depth-four circuits (Saraf & Volkovich 2011).

For their application to multilinear depth-four formulae Karnin *et al.* (2013) consider multilinear formulae of the form $F = \sum_{i=1}^m F_i$

where the F_i's factor into subformulae each depending only on a fraction α of the variables. In such a case we call the formula F α-split [1]. For technical reasons we present a more general definition that requires "splitness" with respect to a restricted set of variables.

Definition 2.18 (α-Split). *Let $F = \sum_{i=1}^{m} F_i \in \mathbb{F}[x_1, \ldots, x_n]$, $\alpha \in [0,1]$, and $V \subseteq [n]$. We say that F is* α-split *if each F_i is of the form $\prod_j F_{i,j}$ where $|\mathrm{var}(F_{i,j})| \leq \alpha n$. F is* α-split with respect to V *(in shorthand, α-split$_V$) if $|\mathrm{var}(F_{i,j}) \cap V| \leq \alpha|V|$ for all i, j.*

For $V = [n]$, the two definitions coincide. Note that in the definition of split we do not require that $\mathrm{var}(F) = [n]$.

To state the structural result Karnin et al. use, we also need the following terminology. An additive top-fanin-m formula $F = \sum_{i=1}^{m} F_i$ is said to be *simple* if the greatest common divisor (gcd) of the F_i's is in $\mathbb{F}$. F is said to be *minimal* if for all non-trivial subsets $S \subsetneq [m]$, $\sum_{i \in S} F_i \not\equiv 0$.

Lemma 2.19 (Structural Witness for Split Multilinear Formulae). *For $R(m) = (m-1)^2$ the following holds for any multilinear formula $F = \sum_{i=1}^{m} F_i$ on $n \geq 1$ variables with $\cup_{i\in[m]}\mathrm{var}(F_i) = [n]$. If F is simple, minimal, and α-split for $\alpha = (R(m))^{-1}$, then $F \not\equiv 0$.*

Although not critical for our results, we point out that Lemma 2.19 shaves off a logarithmic factor in the bound for $R(m)$ obtained by Karnin et al. They show how to transform a split, simple and minimal, multilinear formula $F = \sum_{i=1}^{m} F_i$ into a simple and minimal depth-three formula $F' = \sum_{i=1}^{m} F_i'$, and then apply the so-called rank bound (Dvir & Shpilka 2007; Saxena & Seshadhri 2009, 2010) to F' in order to show that $F \not\equiv 0$. We follow the same outline, but use a new structural witness for the special type of multilinear depth-three formulae F' that arise in the proof, rather than the rank bound.

The special type of multilinear depth-three formulae we consider are of the form $F = \sum_{i=1}^{m} F_i$ where each F_i is the product of univariate linear polynomials. Along the lines of Saraf & Volkovich

[1] Karnin *et al.* (2013) refer to "split" formulae as "compressed".

(2011), we show that such formulae that are simple, minimal, and have a branch that depends on more than $m-2$ variables, cannot be zero. More generally we show that if the greatest common divisor of a non-trivial subset H of the branches depends on more than $m-|H|-1$ variables, then $F \not\equiv 0$ (this reduces to the simpler instantiation we use when $|H|=1$).

Lemma 2.20 (Structural Witness for Univariate Multilinear Depth-Three Formulae). *Let $m \geq 2$ and $F = \sum_{i=1}^{m} F_i = \sum_{i=1}^{m} \prod_{j=1}^{d_i} L_{ij}$ be a multilinear depth-three formula where each L_{ij} is a univariate polynomial. If F is simple, minimal, and there is a nonempty $H \subsetneq \{F_1, \ldots, F_m\}$ with $|\mathrm{var}(\gcd(H))| > m - |H| - 1$, then $F \not\equiv 0$.*

We defer the proof of Lemma 2.20 to Section 2.3.1, and now show how it implies Lemma 2.19.

Proof (of Lemma 2.19). Without loss of generality write $F = \sum_{i=1}^{m} F_i = \sum_{i=1}^{m} \prod_{j=1}^{d_i} P_{ij}$, where the P_{ij} are irreducible. We can construct a set $U \subseteq [n]$ such that for all i,j, $|U \cap \mathrm{var}(P_{ij})| \leq 1$ and there exists $\ell \in [m]$ for which $|U \cap \mathrm{var}(F_\ell)| \geq \frac{1}{\alpha \cdot m} > m-2$.

Construct U greedily as follows. Begin with U empty. While there is a variable x such that all the P_{ij}'s that depend on x depend on no variables currently in U, add x to U. Each added variable x excludes at most $(\alpha n) \cdot b_x$ variables from consideration, where b_x is the number of branches of F that depend on x. This procedure can continue as long as $\sum_{x \in U} \alpha n b_x < n$. This implies that we can achieve $b \doteq \sum_{x \in U} b_x \geq \frac{1}{\alpha}$. Observe that we may also write $b = \sum_{i=1}^{m} |U \cap \mathrm{var}(F_i)|$. By averaging we see that there exists an $\ell \in [m]$ such that $|U \cap \mathrm{var}(F_\ell)| \geq \frac{1}{\alpha \cdot m} > m-2$, as claimed.

Fixing all variables outside of U linearizes each P_{ij} – in fact, each P_{ij} becomes a univariate linear function – and turns F into a depth-three formula F' with an addition gate on top of fanin m. Moreover, as we will argue, a typical assignment $\bar{\beta}$ from $\bar{\mathbb{F}}$ to the variables outside of U keeps F' (1) simple, (2) minimal, and (3) ensures that $|\mathrm{var}(F'_\ell)| > m-2$. The structural witness for univariate multilinear depth-three formulae (Lemma 2.20) with $H = \{F'_\ell\}$ implies that $F' \not\equiv 0$, and therefore that $F \not\equiv 0$.

All that remains is to establish the above claims about a typical assignment $\bar{\beta}$ to the variables in $[n] \setminus U$:

1. To argue simplicity, we make use of the following property of multilinear polynomials P and Q: If P is irreducible and depends on a variable x, then $P|Q$ iff $P|_{x\leftarrow 0} \cdot Q - P \cdot Q|_{x\leftarrow 0} \equiv 0$. The property holds because if $P|Q$, then $Q = P \cdot Q'$ where Q' does not depend on x by multilinearity and hence $P|_{x\leftarrow 0} \cdot Q - P \cdot Q|_{x\leftarrow 0} \equiv P|_{x\leftarrow 0} \cdot (P \cdot Q') - P \cdot (P|_{x\leftarrow 0} \cdot Q') \equiv 0$. If P does not divide Q, then since P is irreducible and depends on x, P does not divide $P|_{x\leftarrow 0} \cdot Q$, and we conclude the required identity cannot hold.

   Since F is simple, for every irreducible subformula P_{ij} that depends on some $u \in U$, there is branch, say $F_{i'}$, such that P_{ij} does not divide $F_{i'}$. Thus, by the above property, $P_{ij}|_{U\leftarrow \bar{0}} \cdot F_{i'} - P_{ij} \cdot F_{i'}|_{U\leftarrow \bar{0}} \not\equiv 0$. Let P'_{ij} be the result of applying $\bar{\beta}$ to P_{ij}, and define $F'_{i'}$ and F' similarly. A typical assignment $\bar{\beta}$ keeps $P_{ij}|_{U\leftarrow 0} \cdot F_{i'} - P_{ij} \cdot F_{i'}|_{U\leftarrow 0}$ non-zero and P'_{ij} dependent on u. Since P'_{ij} remains irreducible as a univariate polynomial, the above property implies that P'_{ij} does not divide $F'_{i'}$. Therefore, F' is simple.

2. Minimality is maintained by a typical assignment since if $\sum_{i\in S} F_i$ is a non-zero polynomial for all $\emptyset \subsetneq S \subsetneq [m]$, then the same holds after a typical partial assignment $\bar{\beta}$.

3. Finally, for any $u \in U$ there exists at least one P_{ij} for which $u \in \text{var}(P_{ij})$. Since a typical assignment to the variables in P_{ij} other than u turns P_{ij} into a non-constant linear function of u and $|U \cap \text{var}(F_\ell)| > m - 2$, we conclude that F'_ℓ depends on more than $m - 2$ variables under a typical assignment $\bar{\beta}$. □

2.3.1. Proof of the Structural Witness Lemma for Split Multilinear Formulae. We now return to the proof of Lemma 2.20, whose outline we briefly sketch. We argue by induction on m. In the base case $m = 2$ we only need to consider singletons H. If F is zero, then the simplicity of F implies that both branches are constants, and thus $\gcd(H)$ does not depend on more than $m - |H| - 1 = 0$ variables. Thus, F has to be zero.

For $m \geq 3$, first assume $h \doteq |H| > 1$. We argue that a typical assignment to the variables outside of $\mathrm{var}(\gcd(H))$ reduces the top fanin of F while maintaining simplicity and minimality (as in the proof of Lemma 2.19) and hence the induction hypothesis implies the required bound on $|\mathrm{var}(\gcd(H))|$. For $h = 1$, we argue by contradiction. Suppose that some branch, say F_m, depends on more than $m-2$ variables but that F is zero. Using the induction hypothesis and the case for $h > 1$ we argue that the branches other than F_m depend on at most $m-2$ variables. Thus, there exists a set of variables $V \subseteq \mathrm{var}(F_m)$ that is not contained in any other branch. The partial derivative of F with respect to V zeroes all branches except F_m. This means that $\partial_V F \equiv \partial_V F_m \not\equiv 0$, contradicting the fact that F is zero, and completing the proof.

Proof (of Lemma 2.20). Consider the base case of $m = 2$. Here H must be a singleton set. If F is simple and zero, since the ring of polynomials over a field is a unique factorization domain, we observe that F_1 and F_2 are constants. Hence each branch depends on at most $m - |H| - 1 = 2 - 1 - 1 = 0$ variables, which contradicts our assumption. Thus F is non-zero.

Consider the induction step for $m \geq 3$. We first assume that $h \doteq |H| > 1$ and without loss of generality let $H \doteq \{F_1, F_2, \ldots, F_h\}$. Factor $F_i = \gcd(H) \cdot f_i$ for $i \in [h]$. We can write

$$F = \gcd(H) \cdot (f_1 + f_2 + \ldots + f_h) + F_{h+1} + \ldots + F_m.$$

As F is multilinear $\mathrm{var}(\gcd(H)) \cap \mathrm{var}(f_i) = \emptyset$ for all $i \in [h]$. As in the proof of Lemma 2.19, we can fix the variables outside $\mathrm{var}(\gcd(H))$ to a typical assignment such that the resulting formula $F' = \gcd(H) \cdot \alpha + F'_{h+1} + \ldots + F'_m$ remains simple, minimal, and zero, and has $\alpha \neq 0$. Since F' has top fanin $m' = m - h + 1 \leq m - 1$, the induction hypothesis (with $h = 1$) implies that $|\mathrm{var}(\gcd(H))| \leq m' - 2 = m - h - 1$. Thus the induction step goes through for $h > 1$.

Now assume that $h = 1$. Suppose without loss of generality that F_m depends on more than $m-2$ variables. Assume by way of contradiction that F is zero. We first show that for all $i \in [m-1]$, $|\mathrm{var}(F_i)| \leq m-2$. Without loss of generality consider F_1.

Since the induction step holds for sets containing at least two branches, $|\mathrm{var}(\gcd(F_1, F_m))| \leq m-3$. Thus there is a univariate factor $(x-\alpha)$ of F_m, for some variable x and constant $\alpha \in \mathbb{F}$, which does not divide F_1. Since $F \equiv 0$, $F|_{x\leftarrow\alpha} \equiv 0$. Observe that $F_m|_{x\leftarrow\alpha} \equiv 0$, but $F_1|_{x\leftarrow\alpha} \not\equiv 0$. Hence there exists a minimal zero subformula F' of $F|_{x\leftarrow\alpha}$ that contains the summand $F_1|_{x\leftarrow\alpha}$. Without loss of generality we write

$$F' \doteq \sum_{i=1}^{h'} F'_i = \sum_{i=1}^{h'} F_i|_{x\leftarrow\alpha},$$

for some $2 \leq h' \leq m-1$. Multilinearity implies that

$$|\mathrm{var}(F_1)| = 1 + |\mathrm{var}(F'_1)| = 1 + \left|\mathrm{var}\left(\frac{F'_1}{\gcd(F')}\right)\right| + |\mathrm{var}(\gcd(F'))|\,.$$

We bound the latter two terms. For the first term, observe that the formula $\frac{F'}{\gcd(F')}$ is simple by construction; in addition, it is minimal because F' is minimal. Consequently, by the induction hypothesis on $\frac{F'}{\gcd(F')}$ we get that $\left|\mathrm{var}\left(\frac{F'_1}{\gcd(F')}\right)\right| \leq h'-2$.

Now consider the second term. We have that

$$|\mathrm{var}(\gcd(F'))| \leq |\mathrm{var}\left(\gcd(F_1, F_2, \ldots, F_{h'})\right)| \leq m-h'-1,$$

where the former inequality follows because the F_i are products of univariate polynomials, and the latter inequality follows because $h' > 1$ and the induction step holds for sets containing more than one branch (but not all branches).

By putting everything together we conclude that

$$|\mathrm{var}(F_1)| \leq 1 + (h'-2) + (m-h'-1) = m-2,$$

and hence that $|\mathrm{var}(F_i)| \leq m-2$ for all $i \in [m-1]$.

Let $V \doteq \cup_{i\in[m-1]}\mathrm{var}(F_m)\backslash\mathrm{var}(F_i)$. Because F_m is a multilinear product of linear univariate polynomials, $\partial_V F_m \not\equiv 0$. However, $\partial_V F_i \equiv 0$ for all $i \in [m-1]$, because $\mathrm{var}(F_m)\backslash\mathrm{var}(F_i) \neq \emptyset$. Hence $\partial_V F \equiv \partial_V F_m \not\equiv 0$, contradicting our hypothesis that $F \equiv 0$. This completes the case for $h = 1$ and the proof. □

3. Fragmenting and Shattering Formulae

In this section we describe a means of splitting up or *fragmenting* multilinear sparse-substituted formulae. We build up towards this goal by first fragmenting read-once formulae, and then multilinear read-k formulae. We conclude by extending our fragmentation technique to work for sparse-substituted formulae, proving our Fragmentation Lemma (Lemma 3.4).

We view the Fragmentation Lemma as an atomic operation that breaks a read-k formula into a product of easier formulae. It does so via a set of carefully chosen partial derivatives and zero-substitutions of the formula; the more such operations are performed the longer the seed length of the eventual generator will be. By greedily applying the Fragmentation Lemma and using some other ideas we are able to *shatter* multilinear sparse-substituted Σ^m-read-k formulae, that is, simultaneously split all the top-level branches so that they are the product of factors that each only depend on a fraction of the variables. The Shattering Lemma (Lemma 3.7) is the main result of this section.

3.1. Fragmenting Read-Once Formulae. We begin by showing that for a non-constant read-once formula F there is a variable $x \in \mathrm{var}(F)$ such that $\partial_x F$ is the product of subformulae of F which each depend on at most half of the variables. Since F is read-once, all gates of F have variable disjoint children, and it suffices to pick x such that the path from the output of F to x bisects F. For technical reasons, we state a more general version of the lemma, namely with respect to a restricted variable set with weights.

Lemma 3.1 (Fragmenting Read-Once Formulae). *Let V be a non-empty set of variables and let F be a multilinear read$_V$-once formula such that $V \subseteq \mathrm{var}(F)$. Let $w : V \to \mathbb{N}$ be a weight function and $W \doteq \sum_{x \in V} w(x)$. There exists a variable $x \in V$ such that $\partial_x F$ is the product of subformulae g of F each of which satisfies $\sum_{y \in V \cap \mathrm{var}(g)} w(y) \leq \frac{W}{2}$.*

Proof. Assume without loss of generality that F has fanin two. Let S be a sequence of the variables in V produced by an in-order tree traversal of the formula F. Such a sequence exists because F

is a read$_V$-once formula. For $x \in V$, let $L(x)$ denote the variables in V occurring before x in S, and $R(x)$ be the variables in V occurring after x in S. Observe that $L(x)$ and $R(x)$ partition $V \backslash \{x\}$. Naturally associate weights with these parts: $W_L(x) \doteq \sum_{y \in L(x)} w(y)$ and $W_R(x) \doteq \sum_{y \in R(x)} w(y)$. There exists an $x \in V$ that is a weighted median of the sequence S, i.e., an $x \in V$ such that $W_L(x), W_R(x) \leq \frac{W}{2}$. Fix such an x, and note that $\partial_x F \not\equiv 0$, because $V \subseteq \mathrm{var}(F)$.

Since F is read$_V$-once, the input gate f labeled x contains all occurrences of x in F. By Proposition 2.4, $\partial_x F = \partial_x f \cdot \prod_{g \in U_F(f)} g = \prod_{g \in U_F(f)} g$, where $U_F(f)$ denotes the unvisited children of the multiplication gates along the path from the output of F to f. Note that for every subformula g of F, the set of variables in V that appear in g form a contiguous subsequence of S because S is an in-order tree traversal. For $g \in U_F(f)$, since F is read$_V$-once and $x \in V$, x does not appear in g and all the variables in V that occur in g must appear entirely on one side of x in the sequence S. Thus $\mathrm{var}(g) \subseteq L(x)$ or $\mathrm{var}(g) \subseteq R(x)$. In either case we conclude that $\sum_{y \in \mathrm{var}(g) \cap V} w(y) \leq \frac{W}{2}$. □

3.2. Fragmenting Multilinear Read-k Formulae. While illustrating the basic idea of fragmenting, Lemma 3.1 is insufficient for our purposes. A key reason the proof of Lemma 3.1 goes through is that in read-once formulae the children of addition gates are variable disjoint. This property implies that there is a unique path from the output gate to any variable. In multilinear read-k formulae this is no longer the case. Our solution is to follow the largest branch that depends on a variable that is only present within that branch. This allows us to mimic the behavior of the read-once approach as long as such a branch exists. Once no such branch exists, each child of the current gate cannot contain all the occurrences of any variable. This means that these children are read-$(k-1)$ formulae. Taking a partial derivative with respect to a variable that only occurs within the current gate eliminates all diverging addition branches above the gate. This makes the resulting formula multiplicative in all the unvisited (and small) multiplication branches. This intuition can be formalized in the

Algorithm 1 – Fragment(g, k, V) – An algorithm fragmenting a read$_V$-k formula g.

```
if $g$ is read$_V$-$(k-1)$ then
    return arbitrary $x \in V$
if $g = g_1 \cdot g_2$, and $\exists\,(i, x) \in \{1, 2\} \times V$ where $x$ occurs $k$ times in $g_i$ and $|\mathrm{var}(g_i) \cap V| > \frac{|V|}{2}$ then
    return Fragment$(g_i, k, V)$
if $g = g_1 + g_2$, and $\exists\,(i, x) \in \{1, 2\} \times V$ where $x$ occurs $k$ times in $g_i$ then
    return Fragment$(g_i, k, V)$
return $x \in V$ that occurs $k$ times in $g$
```

following lemma.

Lemma 3.2 (Fragmenting Multilinear Read-k Formulae). *Let V be a non-empty set of variables, $k \geq 2$, and F be a multilinear read$_V$-k formula such that $V \subseteq \mathrm{var}(F)$. There exists a variable $x \in V$ such that $\partial_x F$ is the product of*

(i) formulae f for which $|\mathrm{var}(f) \cap V| \leq \frac{|V|}{2}$, and

(ii) at most one Σ^2-read$_V$-$(k-1)$ formula.

Moreover, each of these factors is of the form g or $\partial_x g$ where g is a subformula of F.

Proof. Assume without loss of generality that F has fanin two. For clarity we outline our procedure for selecting an appropriate x in Algorithm 1.

If no variable in V occurs k times in F, then F is a Σ^2-read$_V$-$(k-1)$ formula (indeed, it is a read$_V$-$(k-1)$ formula). Hence for any variable $x \in V$ $\partial_x F$ is explicitly a Σ^2-read$_V$-$(k-1)$ formula and the single factor $\partial_x F$ satisfies the required properties. Therefore assume that at least one variable in V occurs k times.

To locate an appropriate variable $x \in V$ we recurse through the structure of F, maintaining the following invariant: There exists a variable $x \in V$ such that the current subformula g being visited contains all k occurrences of x in F. Setting g to be F satisfies this invariant initially.

If the top gate of g is a multiplication gate, recurse on the child that depends on more than $\frac{|V|}{2}$ of the variables in V and contains k occurrences of some variable in V. If no such child exists, end the recursion at g and select a variable from V that occurs k times in g. Such a variable must exist by the invariant.

If the top gate of g is an addition gate, $g = g_1 + g_2$, and at least one of its children, g_i, has a variable in V that occurs k times in g_i, recurse on g_i. Otherwise, both children of g are read$_V$-$(k-1)$ formulae. Select a variable from V that occurs k times in g ending the recursion. Again, such a variable must exist by the invariant.

Let $x \in V$ be the variable selected by the procedure. We argue that $\partial_x F$ can be written in the desired form. Denote by f the subformula where the recursion ended. Since f contains all the occurrences of x in F, Proposition 2.4 tells us that $\partial_x F = \partial_x f \cdot \prod_{g \in U_F(f)} g$, where $U_F(f)$ denotes the unvisited multiplication gates on the path from the root of F to f. By the selection rule for multiplication gates, all the gates $g \in U_F(f)$ each depend on at most $\frac{|V|}{2}$ variables from V. All that remains is to analyze $\partial_x f$. There are two cases depending on the top gate of f.

Suppose the top gate of f is a multiplication gate: $f = f_1 \cdot f_2$. Without loss of generality assume that $x \in \mathrm{var}(f_1)$. Since F is multilinear, we can write

$$\partial_x f = (\partial_x f_1) \cdot f_2. \tag{3.3}$$

The stopping rule and multilinearity together imply that f_1 (and hence $\partial_x f_1$ depends on at most $\frac{|V|}{2}$ of the variables in V, and that f_2 either:

(i) depends on at most $\frac{|V|}{2}$ of the variables in V, or

(ii) does not contain k occurrences of any variable in V and therefore is a Σ^2-read$_V$-$(k-1)$ formula.

In either case, the resulting factoring of $\partial_x F$ satisfies the properties in the statement of the lemma.

Suppose the top gate of f is an addition gate: $f = f_1 + f_2$. According to the stopping rule both children of f are read$_V$-$(k-1)$ formulae, making f a Σ^2-read$_V$-$(k-1)$ formula, and so is $\partial_x f$. The resulting factoring of $\partial_x F$ again satisfies the properties in the statement of the lemma. □

For the case $k = 1$ the proof of Lemma 3.2 yields the unweighted version of Lemma 3.1.

3.3. Fragmenting Sparse-Substituted Formulae. In this subsection we extend our fragmenting arguments to work for sparse-substituted formulae.

First consider a multilinear sparse-substituted read-once formula F. The idea is to apply the argument from Lemma 3.1 and the chain rule to locate a variable x such that $\partial_x F$ is *almost* fragmented. By this we mean that each of the factors of $\partial_x F$ depends on at most half of the variables except the factor that was originally a sparse polynomial that depends on x. The sparse polynomial, say f, may depend on too many variables. In that case we perform further operations so that f factors into smaller pieces. Through a sequence of partial derivatives and zero-substitutions we eliminate all but one term in f. This implies that the sparse polynomial and hence the overall resulting formula F' is $\frac{1}{2}$-split. To perform the additional step, observe that for any variable x, either at most half of the terms in f depend on x or at most half do not. In the former case, taking the partial derivative with respect to x eliminates at least half of the terms; setting x to 0 has the same effect in the latter case. Repeating this process a number of times logarithmic in the maximum number of terms eliminates all but one of the terms, resulting in a trivially split formula.

This is the intuition behind the sparse-substituted extension of Lemma 3.1 and corresponds to the first part of the next lemma. The second part is the sparse-substituted extension of Lemma 3.2 and follows from that lemma by a simple observation.

Lemma 3.4 (Fragmentation Lemma). *Let V be a non-empty set of variables, $k \geq 1$, and F be a multilinear sparse-substituted read$_V$-k formula such that $V \subseteq \mathrm{var}(F)$. Let t denote the maximum* *number of terms in each substituted polynomial.*

(i) *If $k = 1$, there exist disjoint sets of variables P, Z with $|P \cup Z| \leq \log(t) + 1$ such that $\partial_P F|_{Z \leftarrow 0}$ is non-zero and is a product of factors f for which $|\mathrm{var}(f) \cap V| \leq \frac{|V|}{2}$. Moreover, the factors are subformulae of F.*

(ii) *If $k \geq 2$, there exists a variable $x \in V$ such that $\partial_x F$ is the product of*

   (a) *formulae f for which $|\mathrm{var}(f) \cap V| \leq \frac{|V|}{2}$, and*

   (b) *at most one Σ^2-read$_V$-$(k-1)$ formula.*

   Moreover, each of these factors is of the form g or $\partial_x g$ where g is a subformula of F.

Proof. We argue the two parts separately.

Part 1. Assume without loss of generality that F has fanin two.

Write $F = B(\rho_1, \ldots, \rho_r)$ as in Definition 2.3, where $B \in \mathbb{F}[Y]$ is the backbone of F and the ρ_i's are t-sparse multilinear polynomials. Define $Y' \doteq \{y_i \in Y \mid |\mathrm{var}(\rho_i) \cap V| > 0\}$ and the weight function $w : Y' \to \mathbb{N}$ by $y_i \mapsto |\mathrm{var}(\rho_i) \cap V|$. Because F is a read$_V$-once formula and $V \subseteq \mathrm{var}(F)$, $W \doteq \sum_{y \in Y'} w(y) = |V|$. Apply Lemma 3.1 to B with set Y' and weight function w to determine a variable $y_i \in Y' \subseteq Y$ such that we can write $\partial_{y_i} B = \prod_j b_j$ where the b_j's are subformulae of B and $\sum_{y \in \mathrm{var}(b_j)} w(y) \leq \frac{W}{2} = \frac{|V|}{2}$. Pick $x \in \mathrm{var}(\rho_i) \cap V$. Note that x exists because $i \in Y'$. Since F is a read$_V$-once formula, only ρ_i depends on x and applying the chain rule produces

$$\begin{aligned}\partial_x F = \partial_x B(\rho_1, \ldots, \rho_r) &= (\partial_x \rho_i) \cdot (\partial_{y_i} B(y_1, \ldots, y_r))|_{Y \leftarrow \bar{\rho}} \\ &= (\partial_x \rho_i) \cdot \prod\nolimits_j g_j\end{aligned}$$

where $g_j \doteq b_j|_{Y \leftarrow \bar{\rho}}$. Observe that each g_j is a subformula of F and depends on at most $\frac{|V|}{2}$ variables from V. If $\partial_x \rho_i$ depends on at most $\frac{|V|}{2}$ of the variables in V, the lemma is complete with $P \doteq \{x\}$ and $Z \doteq \emptyset$. Therefore, assume otherwise.

Let $\rho \doteq \partial_x \rho_i$. Write $\rho = \sum_{j=1}^{\ell} M_j$ where $\ell \leq t$ and each M_j is a distinct term.

Claim 3.5. *There exist disjoint $P', Z \subseteq \mathrm{var}(\rho)$ such that $|P' \cup Z| \leq \log \ell$ and $\partial_{P'}\rho|_{Z\leftarrow 0}$ is a single term.*

Proof. We proceed by induction on ℓ. In the base case $\ell = 1$. Trivially, ρ is a term. It suffices to set $P' = Z = \emptyset$ and hence $|P' \cup Z| = 0 = \log 1$.

In the induction step $\ell > 1$, there is a variable $z \in \mathrm{var}(\rho)$ that divides some term M_j of ρ, but not all terms of ρ. To see this, suppose that each variable in $\mathrm{var}(\rho)$ divides each term of ρ, then, because ρ is multilinear, each term must be the same, so $\ell = 1$. This contradicts $\ell > 1$. We complete by considering two cases based on the number of terms that z divides. Suppose z divides at most $\frac{\ell}{2}$ terms of ρ. Then $\partial_z\rho$ has at most $\frac{\ell}{2}$ terms and is non-zero. By induction there are sets $P'', Z' \subseteq \mathrm{var}(\partial_z\rho) \subseteq \mathrm{var}(\rho)$ with $|P'' \cup Z'| \leq \log \frac{\ell}{2} = (\log \ell) - 1$ such that $\partial_{P''}\partial_z\rho|_{Z'\leftarrow 0}$ is a single term. Set $P' \doteq P'' \cup \{z\}$ and $Z \doteq Z'$ to conclude this case. Otherwise, z divides more than $\frac{\ell}{2}$ terms of ρ, but not all of them. Thus $\rho|_{z\leftarrow 0}$ has at most $\frac{\ell}{2}$ terms and is non-zero. Applying the induction hypothesis to $\rho|_{z\leftarrow 0}$ suffices to conclude as in the previous case but setting $P' \doteq P''$ and $Z \doteq Z' \cup \{x\}$. □

Apply Claim 3.5 to ρ to get $P' \cup Z \subseteq \mathrm{var}(\rho)$ with $|P' \cup Z| \leq \log t$. Set $P \doteq P' \cup \{x\}$. Note that $\partial_P F|_{Z\leftarrow 0} = (\partial_P \rho_i|_{Z\leftarrow 0}) \cdot \prod_j g_j \not\equiv 0$, because F is multilinear, $P \cup Z \subseteq \mathrm{var}(\rho)$, and $\partial_P\rho_i|_{Z\leftarrow 0}$ is a non-zero term. Finally, observe that because $\partial_P\rho_i|_{Z\leftarrow 0}$ is a term it trivially factors into a product of subformulae of F (namely variables) that each depend on at most one variables in V. We conclude by noting that $1 \leq \frac{|V|}{2}$. This follows because ρ_i depends on $x \in V$ and at least one other variable in V (since $\partial_x\rho_i$ depends on more than $\frac{|V|}{2}$ variables in V other than x), hence $|V| \geq 2$.

Part 2. Here the proof is essentially the same as the proof of Lemma 3.2. Since $k \geq 2$, the argument always halts at an internal gate and never reaches a sparse-substituted input. Only the number of occurrences of each variable is relevant to the decisions the argument makes. This implies that the argument does not change when sparse-substituted formulae are considered (and is even independent of the sparsity parameter). Thus, this part of the proof is immediate as a corollary to the proof of Lemma 3.2. □

Observe that the cost of applying the Fragmentation Lemma to a read-once formula is $\log(t) + 1$ partial derivatives and zero-substitutions, whereas applying it to a formula that is not read-once requires only a single partial derivative (though the promised result is weaker in this case).

It is useful to have a version of Part 2 of the Fragmentation Lemma generalized to structurally-multilinear formulae. The argument is the same as for the earlier version except that in addition to selecting an appropriate x, we must pick an $\alpha \in \mathbb{F}$, such that $\partial_{x,\alpha} F \doteq F|_{x \leftarrow \alpha} - F|_{x \leftarrow 0}$ is non-zero. The directed partial derivative comes in here because $\partial_x F \doteq F|_{x \leftarrow 1} - F|_{x \leftarrow 0}$ may be zero even when F depends on x, because F is not multilinear.

Lemma 3.6 (Fragmentation Lemma for Structurally-Multilinear Formulae). *Let V be a non-empty set of variables, $k \geq 2$, and F be a structurally-multilinear sparse-substituted read$_V$-k formula with $V \subseteq \mathrm{var}(F)$. Let t denote the maximum number of terms in each substituted polynomial. There exists a variable $x \in V$ and $\alpha \in \bar{\mathbb{F}}$ such that $\partial_{x,\alpha} F$ is non-zero and the product of*

(i) formulae f for which $|\mathrm{var}(f) \cap V| \leq \frac{|V|}{2}$, and

(ii) at most one Σ^2-read$_V$-$(k-1)$ formula.

Moreover, each of these factors is of the form g or $\partial_x g$ where g is a subformula of F.

Proof. Repeat the proof of Lemma 3.4, Part 2, but add the following step. After selecting an appropriate variable x that F depends on, select an α such that $\partial_{x,\alpha} F \not\equiv 0$. By the Schwartz-Zippel Lemma, such an α exists within the algebraic closure $\bar{\mathbb{F}}$ of $\mathbb{F}$. Note that, in fact, if $|\mathbb{F}|$ is larger than the degree of x in F such an α is present in $\mathbb{F}$. □

3.4. Shattering Multilinear Formulae. The previous subsections establish a method for fragmenting multilinear sparse-substituted read-k formulae. We now apply the Fragmentation Lemma (Lemma 3.4) to simultaneously split all branches of a multilinear sparse-substituted Σ^m-read-k formula. We call this process *shattering*. When $k = 1$, applying the Fragmentation Lemma greedily to a factor of a branch that depends on the largest number of variables suffices to split a multilinear sparse-substituted Σ^m-read-once formula to an arbitrary level. To obtain an α-split formula in the end, we need $O(m^{\frac{(\log(t)+1)}{\alpha}})$ partial derivatives and zero-substitutions.

In the case of arbitrary read-value $k > 1$ the Fragmentation Lemma is not immediately sufficient for the task. As in the read-once case, we can apply the lemma greedily to a largest factor of a read-k branch to α-split the branch within at most $\frac{2}{\alpha}$ applications. However, this is assuming that Case (ii.b) of the Fragmentation Lemma never occurs where the Σ^2-read-$(k-1)$ factor depends on more than half (possibly all) of the variables. When this case occurs the fragmentation process fails to split the formula into pieces each depending on few variables. To resolve the issue, we leverage the fact that this troublesome factor is both large and a Σ^2-read-$(k-1)$ formula.

Consider a read-k formula F on n variables. Apply the Fragmentation Lemma to F. Suppose that Case (ii.b) of the lemma occurs, producing a variable x, and that the corresponding Σ^2-read-$(k-1)$ factor of $\partial_x F$ depends on more than $\frac{n}{2}$ of the variables. Without loss of generality, $\partial_x F = H \cdot (H_1 + H_2)$, where H is a product of read-k formulae each depending on at most $\frac{n}{2}$ variables, and both H_1 and H_2 are read-$(k-1)$ formulae. Rewrite F by distributing the top level multiplication over addition:

$$F' \doteq (H \cdot H_1) + (H \cdot H_2) \equiv H \cdot (H_1 + H_2) = \partial_x F.$$

Let $V \doteq \mathrm{var}(H_1 + H_2)$. F' is explicitly a Σ^2-read$_V$-$(k-1)$ formula and a read$_V$-k formula. However, F' is almost certainly not a read-k formula. Partition the variables into "same-read" sets, that is, sets of variables that appear the exact same number of times. By further restricting to the largest "same-read" set of variables in the larger of the two subformulae H_1 and H_2, we can argue the existence of a subset $V' \subseteq V$ that contains at least a $\frac{1}{2k}$ fraction of the variables in V such that the read of H_1 and H_2 with respect to V' sum to at most k. Note that prior to this restriction the upper bound on this sum is $2(k-1)$. This action effectively breaks up the original formula F into two branches without increasing the sum of the read values of the branches. Since $|V| \geq \frac{n}{2}$, the set V' is at most a factor $4k$ smaller than n, and the number of branches increased by one.

This operation can be performed at most $k-1$ times on a read-k formula before either: (i) the attempted greedy splitting is successful, or (ii) the formula becomes the sum of k read-once formulae with respect to some subset V of $[n]$. In the latter case we are effectively in the situation we first described with $k=1$, and all subsequent splittings will succeed. In either case we obtain a formula that is shattered with respect to a subset V that is at most a factor $k^{O(k)}$ smaller than n.

In summary, the Shattering Lemma splits multilinear sparse-substituted Σ^m-read-k formulae to an arbitrary extent, albeit with some restriction of the variable set and an increase in top fanin. Moreover, each of the branches in the shattered formula are present in the original input formula, either as such or after taking some partial derivatives and zero-substitutions. This technical property follows from the properties of the Fragmentation Lemma and will be needed in the eventual application.

Lemma 3.7 (Shattering Lemma). *Let $\alpha : \mathbb{N} \to (0,1]$ be a non-increasing function. Let $F \in \mathbb{F}[x_1, \ldots, x_n]$ be a formula of the form $F = c + \sum_{i=1}^{m} F_i$, where c is a constant, and each F_i is a non-constant multilinear sparse-substituted read-k_i formula. Let t denote the maximum number of terms in each substituted polynomial. There exist disjoint subsets $P, Z, V \subseteq [n]$ such that $\partial_P F|_{Z \leftarrow 0}$ can be written as $c' + \sum_{i=1}^{m'} F'_i$, where c' is a constant, and*

- $m' \leq k \doteq \sum_{i=1}^{m} k_i$,
- *each F'_i is multilinear and $\alpha(m'+2)$-split$_V$,*
- $|P \cup Z| \leq (k-m+1) \cdot \frac{4k}{\alpha(k+2)} \cdot (\log(t)+1)$, *and*
- $|V| \geq \left(\frac{\alpha(k+2)}{8k}\right)^{k-m} \cdot n - \frac{8k}{\alpha(k+2)} \cdot (\log(t)+1)$.

Moreover, the factors of each of the F'_i's are of the form $\partial_{\tilde{P}} f|_{Z \leftarrow 0}$, where f is some subformula of some F_j and $\tilde{P} \subseteq P$.

Proof. We iteratively construct disjoint subsets $P, Z, V \subseteq [n]$, maintaining the invariant that $\partial_P F|_{Z \leftarrow 0}$ can be written as $F' \doteq c' + \sum_{i=1}^{m'} F'_i$ where

(1) each F'_i is a read$_V$-k'_i formula and c' is a constant

(2) $m' \leq k$

(3) $\sum_{i=1}^{m'} k'_i \doteq k' \leq k$

(4) each F'_i is the product of factors of the form $\partial_{\tilde{P}} f|_{Z \leftarrow 0}$ where f is some subformula of some F_j and $\tilde{P} \subseteq P$.

Setting $P \leftarrow \emptyset, Z \leftarrow \emptyset, V \leftarrow [n]$ and $F' \leftarrow F$ realizes the invariant initially. The fact that $m' \leq k$ follows because each F_i is non-constant.

The goal of our algorithm is to $\alpha(m'+2)$-split$_V$ the formula F'. Each iteration (but the last) consists of two phases: a splitting phase, and a rewriting phase. In the splitting phase we attempt to split F' by greedily applying the Fragmentation Lemma (Lemma 3.4) on each of the branches F'_i. The splitting phase may get stuck because of a Σ^2-read$_V$-$(k'_i - 1)$ subformula that blocks further splitting. If not and the resulting F' is sufficiently split, the algorithm halts. Otherwise, the algorithm enters the rewriting phase where it expands the subformula that blocked the Fragmentation Lemma and reasserts the invariant, after which the next iteration starts. A potential argument shows that the number of iterations until a successful splitting phase is bounded by $k - m$. We first describe the splitting and rewriting phases in more detail, then argue termination and analyze what bounds we obtain for the sizes of the sets P, Z, and V.

Splitting. Assume that F' is not $\frac{\alpha(m'+2)}{2}$-split$_V$, otherwise halt. Let F'_{ij} be a subformula of F' that depends on the most variables in V out of all the factors of the F'_i's. Apply the Fragmentation Lemma (Lemma 3.4) with respect to the set $V \cap \text{var}(F'_{ij})$ to produce sets $P', Z' \subseteq [n]$. Since F' is multilinear, the Fragmentation Lemma implies one of the following holds:

(i) The factors of $\partial_{P'} F'_{ij}|_{Z' \leftarrow 0}$ depend on at most $\frac{|V \cap \mathrm{var}(F'_{ij})|}{2}$ variables in $V \cap \mathrm{var}(F'_{ij})$, or

(ii) $\partial_{P'} F'_{ij}|_{Z' \leftarrow 0}$ has one multilinear sparse-substituted Σ^2-read$_V$-$(k'_i - 1)$ factor which depends on more than $\frac{|V \cap \mathrm{var}(F'_{ij})|}{2}$ variables in $V \cap \mathrm{var}(F'_{ij})$.

Repeatedly perform this greedy application, adding elements to the sets P' and Z' until either case (ii) above occurs or $\partial_{P'} F'|_{Z' \leftarrow 0}$ is $\frac{\alpha(m'+2)}{2}$-split$_V$. In the former case we start a rewriting phase and modify $\partial_{P'} F'|_{Z' \leftarrow 0}$ before we re-attempt to split. In the latter case our goal has been achieved provided that $|P' \cup Z'| \leq \frac{|V|}{2}$: We can add P' to the set P we already had, similarly add Z' to Z, and replace V by $V' \doteq V \setminus (P' \cup Z')$. The assumption that $|P' \cup Z'| \leq \frac{|V|}{2}$ guarantees that $|V'| \geq \frac{|V|}{2}$. Since $\partial_P F|_{Z \leftarrow 0}$ (which equals $\partial_{P'} F'|_{Z' \leftarrow 0}$) is $\frac{\alpha(m'+2)}{2}$-split$_V$, the latter inequality implies that the formula is $\alpha(m'+2)$-split$_{V'}$. If the assumption that $|P' \cup Z'| \leq \frac{|V|}{2}$ does not hold, then outputting $V' = \emptyset$ will meet the size bound for that set and trivially make the formula $\partial_P F|_{Z \leftarrow 0}$ $\alpha(m'+2)$-split$_{V'}$.

The splitting phase maintains the invariant. Regarding Part (4) of the invariant, observe that the factors produced by the Fragmentation Lemma are subformulae of the input to the Fragmentation Lemma (for which the invariant initially held).

Rewriting. We now describe the rewriting phase. Consider the set of variables V at the beginning of the preceding splitting phase. Let F'_{ij} be the subformula the splitting phase blocked on, and let H_1 and H_2 denote the two branches of the multilinear sparse-substituted Σ^2-read$_V$-$(k'_i - 1)$ subformula of $\partial_x F'_i$ that caused the blocking Case (ii.b) of the Fragmentation Lemma to happen. We have that $\partial_{P'} F'_{ij}|_{Z' \leftarrow 0} = H \cdot (H_1 + H_2)$, where H is some read$_V$-k'_i formula that is independent of the variables in $V \cap \mathrm{var}(H_1 + H_2)$. Let $V' \doteq V \cap \mathrm{var}(H_1 + H_2)$. Partition V' into sets $\{V'_0, ..., V'_{k'_i - 1}\}$ based on the exact number of occurrences of each variable in H_1. Let V'' be any set from this partitioning excluding the set V'_0 (we will restrict the choice of V'' later). This implies that H_1 is read$_{V''}$-

k'_{i1} and H_2 is read$_{V''}$-k'_{i2} for some integers k'_{i1} and k'_{i2} such that $k'_{i1}, k'_{i2} < k'_i$ and $k'_{i1} + k'_{i2} \leq k'_i$.

Rewrite $\partial_{P'} F'|_{Z' \leftarrow 0}$ as a top fanin $m'+2$ formula by distributing multiplication over addition in the term $\partial_{P'} F'_i|_{Z' \leftarrow 0}$:

$$\partial_{P'} F'|_{Z' \leftarrow 0} \equiv (H \cdot H_1) + (H \cdot H_2) + \sum_{j \neq i} \partial_{P'} F'_j|_{Z' \leftarrow 0}. \tag{3.8}$$

Observe that $\sum_{j \neq i} \partial_{P'} F'_j|_{Z' \leftarrow 0}$ is a read$_{V''}$-$(\sum_{j \neq i} k'_j)$ formula as partial derivatives and substitutions do not increase the read-value, and $V'' \subseteq V$. The term $(H \cdot H_1) + (H \cdot H_2)$ may not be a read$_V$-k'_i formula, but it must be a read$_{V'}$-k'_i formula. It is explicitly the sum of a read$_{V''}$-k'_{i1} formula and a read$_{V''}$-k'_{i2} formula for some $k'_{i1}, k'_{i2} < k'_i$ with $k'_{i1} + k'_{i2} \leq k'_i$. The representation of $\partial_{P'} F'|_{Z' \leftarrow 0}$ in Equation (3.8) is therefore a read$_{V''}$-k' formula with top fanin $m' + 2$.

Set F' to be this representation of $\partial_{P'} F'|_{Z' \leftarrow 0}$. Merge branches that have become constant into a single constant branch. This maintains the invariant that $m' \leq k' \leq k$. Setting $V \leftarrow V''$ makes F' a top-fanin-$(m' + 1)$ read$_V$-k' formula. As for Part (4) of the invariant, note that the subformula F'_{ij} which blocked the Fragmentation Lemma originally satisfied it during the splitting phase. This means that with respect to the additional partial derivatives and zero-substitutions performed for the attempted split, H_1 and H_2, as well as H, satisfy the invariant as new factors of the branches F'_i. Thus, the new F' satisfies the full invariant. This completes the rewriting phase and one full iteration of the algorithm.

Correctness. We repeat the sequence of splitting and rewriting phases until a splitting phase runs till completion. In that case the algorithm produces disjoint sets $P, Z, V \subseteq [n]$ such that $\partial_P F|_{Z \leftarrow 0}$ can be written as a $\alpha(m'+2)$-split$_V$ formula with top fanin $m'+1 \leq k + 1$.

Apart from the size bounds on the sets P, Z, and V, all that remains to establish correctness is termination. To argue the latter we use the following potential argument. Consider the sum $\sum_{i=1}^{m'} k'_i$ and view it as m' blocks of integer size, where k'_i is the size of the ith block. Over the course of the algorithm blocks can only stay the same, shrink, or be split in a nontrivial way. The latter is what happens in a rewriting phase. As soon as all blocks are of size at most 1, the splitting phase is guaranteed to run successfully because Case (ii.b) of the Fragmentation Lemma cannot occur for read-once formulae, and the algorithm terminates. As we start out with m nontrivial blocks and a value of k for the sum, there can be no more than $k - m$ nontrivial splits. Therefore, there are no more than $k - m$ rewriting phases and $k - m + 1$ splitting phases.

Analysis. We now bound the size of $P \cup Z$. We first analyze how many times the Fragmentation Lemma is applied in each splitting phase. The goal is to $\frac{\alpha(m'+2)}{2}$-split$_V$ each of the m' branches. To $\frac{\alpha(m'+2)}{2}$-split$_V$ one branch, $\frac{4}{\alpha(m'+2)}$ applications of the Fragmentation Lemma are sufficient, since each application reduces the intersection of the factors with V to at most half the original amount. Since the invariant maintains $m' \leq k$ and α is non-increasing, we can upper bound the number of applications of the Fragmentation Lemma during an arbitrary iteration by $\frac{4m'}{\alpha(m'+2)} \leq \frac{4k}{\alpha(k+2)}$. Each single application of the Fragmentation Lemma adds at most $(\log(t) + 1)$ variables to P' and Z'. Since there are at most $k - m + 1$ splitting phases, across all iterations at most $(k - m + 1)(\log(t) + 1)\frac{4k}{\alpha(k+2)}$ variables are added to $P \cup Z$.

We finish by lower bounding the size of V. Consider the change in $|V|$ over one combined splitting/rewriting iteration. We have that $|V'| \geq \frac{\alpha(m'+2)}{4}|V|$, because F' was not $\frac{\alpha(m'+2)}{2}$-split$_V$ before attempting to split F'_{ij} (so the largest number of variables in V that a factor depends on is at least $\frac{\alpha(m'+2)}{2} \cdot |V|$), F'_{ij} was chosen for its maximal dependence on variables from V, and $|\text{var}(H_1 + H_2) \cap V| \geq |\text{var}(F'_{ij}) \cap V|/2$. If we pick V'' to be the largest set from the partitioning $\{V'_0, V'_1, ..., V'_{k'_i-1}\}$ excluding V'_0, and we assume without loss of generality that $|\text{var}(H_1)| \geq |\text{var}(H_2)|$, we have that $|V''| \geq \frac{1}{2(k'_i-1)}|V'|$. Combining these inequalities and using the facts that α is non-increasing and $k \geq k'_i, m'$ gives:

$$|V''| \geq \frac{1}{2(k'_i - 1)}|V'| \geq \frac{\alpha(m' + 2)}{8(k'_i - 1)}|V| \geq \frac{\alpha(k + 2)}{8k}|V|.$$

This means that $|V|$ decreases by a factor of at most $\frac{8k}{\alpha(k+2)}$ in each combined splitting/rewriting iteration. At the end of the final splitting phase $|V'| \geq |V|-2|P' \cup Z'|$ because V' is set to the empty set when $|P' \cup Z'| \geq \frac{|V|}{2}$. Recall that $|P' \cup Z'| \leq \frac{4k}{\alpha(k+2)}(\log(t)+1)$. Since there are at most $k-m$ combined splitting/rewriting iterations, this gives the following lower bound at the end:

$$|V| \geq \left(\frac{\alpha(k+2)}{8k}\right)^{k-m} \cdot n - \frac{8k}{\alpha(k+2)} \cdot (\log(t)+1).$$

□

4. Reducing Testing Read-$(k+1)$Formulae to Testing Σ^2-Read-k Formulae

In this section we describe two methods of reducing identity testing structurally-multilinear sparse-substituted read-$(k+1)$ formulae to identity testing structurally-multilinear sparse-substituted Σ^2-read-k formulae. The first reduction is non-blackbox and is elementary. The second reduction is blackbox and makes use of the Fragmentation Lemma of the preceding section.

4.1. Non-Blackbox Reduction. In the non-blackbox setting we only need to deal with multilinear sparse-substituted formulae, because we can efficiently transform structurally-multilinear sparse-substituted read-k formulae into multilinear sparse-substituted read-k formulae in a non-blackbox way while preserving non-zeroness using the transformation $\mathcal{L}$ from Section 2.1.4 (see Definition 2.7). Recall Section 1.2.2 for the intuition behind the following non-blackbox reduction from identity testing multilinear sparse-substituted read-$(k+1)$ formulae to identity testing multilinear sparse-substituted Σ^2-read-k formulae.

Lemma 4.1 (Read-$(k+1)$ PIT $\leq \Sigma^2$-Read-k PIT – Non-Blackbox Multilinear). *For an integer $k \geq 0$, given a deterministic identity test for n-variate size-s multilinear sparse-substituted Σ^2-read-k formulae that runs in time $T(k,n,s,t)$, where t denotes the maximum number of terms in each substituted polynomial, there is a deterministic algorithm that tests n-variate size-s multilinear sparse-substituted read-$(k+1)$ formulae that runs in time $O((k+1)n \cdot T(k,n,s,t) + \mathrm{poly}(s))$.*

Algorithm 2 – $\textsc{Reduce}_k(g)$

```
if $g = g_1$ op $g_2$ then
    $g \leftarrow \textsc{Reduce}_k(g_1)$ op $\textsc{Reduce}_k(g_2)$
for all variables $x$ appearing in $g$:
    if $x \notin \mathrm{var}(g)$ then
        $g \leftarrow g|_{x \leftarrow 0}$
    else if $g = g_1$ op $g_2$ and $x$ appears $k+1$ times in $g$ then
        return a fresh variable $y_g$
return $g$
```

Proof. Consider the algorithm $\textsc{Reduce}_k$ described in Algorithm 2. Let F be a multilinear sparse-substituted read-$(k+1)$ formula. We first argue that computing $\textsc{Reduce}_k(F)$ suffices to test F, then prove several properties of $\textsc{Reduce}_k$, and conclude by describing how to efficiently compute $\textsc{Reduce}_k(F)$ using the given algorithm for multilinear sparse-substituted Σ^2-read-k formulae.

Let g be a gate of F. Define $F|_{g \leftarrow g'}$ to be the formula resulting from F by replacing the gate g with another formula g'.

Claim 4.2. *For every formula g:*

(i) $\textsc{Reduce}_k(g)$ *is a multilinear sparse-substituted read-*$\max(k, 1)$ *formula.*

(ii) For every variable x appearing in $\textsc{Reduce}_k(g)$, $x \in \mathrm{var}(\textsc{Reduce}_k(g))$.

(iii) For every multilinear sparse-substituted read-$(k+1)$ formula F containing g as a gate, $F \equiv 0$ iff $F|_{g \leftarrow \textsc{Reduce}_k(g)} \equiv 0$.

Applied at the output gate, property (iii) implies that $F \equiv 0$ iff $\textsc{Reduce}_k(F) \equiv 0$. If $\textsc{Reduce}_k(F)$ contains a variable, then by property (ii) $\textsc{Reduce}_k(F)$ must be non-constant and thus nonzero. If $\textsc{Reduce}_k(F)$ contains no variables, then it is constant and it suffices to evaluate the formula to determine whether that constant is zero. Thus, to identity test F it suffices to compute and then examine $\textsc{Reduce}_k(F)$, and the latter examination can be done in time $\mathrm{poly}(s)$.

We argue Claim 4.2 by structural induction on g. In the base case, g is a sparse-substituted input. Property (i) holds as the sparse-substituted inputs are defined to be read-once. Property (ii) holds because the only action of $\textsc{Reduce}_k$ is to eliminate all variables that g does not depend on. It immediately follows that $F \equiv F|_{g \leftarrow \textsc{Reduce}_k(g)}$ and property (iii) holds.

In the induction step, $g = g_1 \operatorname{op} g_2$ for $\operatorname{op} \in \{+, \times\}$. Applying the induction hypothesis twice we have that $F \equiv 0$ iff $F|_{g_1 \leftarrow \textsc{Reduce}_k(g_1)} \equiv 0$ iff

$$\begin{aligned}&(F|_{g_1 \leftarrow \textsc{Reduce}_k(g_1)})|_{g_2 \leftarrow \textsc{Reduce}_k(g_2)}\\ \equiv\ & F|_{g \leftarrow (\textsc{Reduce}_k(g_1) \operatorname{op} \textsc{Reduce}_k(g_2))} \equiv 0.\end{aligned}$$

There are two cases.

1. After reducing the children of g, there is a variable $x \in \operatorname{var}(g)$ that appears $k+1$ times in g.

   In that case, $\textsc{Reduce}_k(g)$ returns y_g. Since y_g is a read-once formula, properties (i) and (ii) are immediate. We now argue property (iii), i.e., that $F \equiv 0$ iff $F|_{g \leftarrow y_g} \equiv 0$.

   Since F is a read-$(k+1)$ formula, g must contain every occurrence of x. Thus $F|_{g \leftarrow y_g}$ does not depend on x. As y_g occurs only once in $F|_{g \leftarrow y_g}$, without loss of generality, write: $F|_{g \leftarrow y_g} \equiv P + Q \cdot y_g$, for two polynomials P and Q that do not depend on x or y_g. If $Q \equiv 0$, then F is independent of g, so $F|_{g \leftarrow y_g} \equiv F$. If $Q \not\equiv 0$, then F is non-zero because $Q \cdot g$ depends on x but P does not, and $F|_{g \leftarrow y_g} \not\equiv 0$ for a similar reason. In both cases we conclude that $F \equiv 0$ iff $F|_{g \leftarrow y_g} \equiv 0$ and property (iii) holds.

2. After reducing the children every variable $x \in \operatorname{var}(g)$ occurs at most k times in g.

   In this case, $\textsc{Reduce}_k(g)$ is explicitly a read-k formula and property (i) holds. Properties (ii) and (iii) hold because the loop eliminates each variable $x \notin \operatorname{var}(g)$ that occurs in g while not changing the polynomial computed at g.

This finishes the proof of Claim 4.2. The correctness of the overall algorithm follows by applying Claim 4.2 with $g = F$.

It remains to argue that $\textsc{Reduce}_k(F)$ can be efficiently computed. An inspection of $\textsc{Reduce}_k$ shows that it does not increase the size, number variables, or sparsity of gates it transforms. The main difficulty is implementing the test $x \notin \mathrm{var}(g)$ from Line 4.

1. When g is a sparse polynomial, $x \notin \mathrm{var}(g)$ can be decided in $\mathrm{poly}(s)$ time by summing coefficients of identical terms in g's list of monomials to aggregate duplicates and then searching for a monomial with non-zero coefficient in which x appears.

2. Otherwise g is an internal gate. Property (ii) for the reduced children of g implies that they depend on exactly those variables that occur in them.

    (a) When g is a multiplication gate and both reduced children of g contain variables, g depends on exactly those variables that appear in g and hence the test in Line 4 is always false. When g is a multiplication gate and at least one reduced child contains no variables, determine the constant value of those children. If a constant child evaluates to zero, $g \equiv 0$ and the test in Line 4 is always true, otherwise it is always false. Thus when g is a multiplication gate, $x \notin \mathrm{var}(g)$ can be decided in $\mathrm{poly}(s)$ time.

    (b) When g is an addition gate and at least one reduced child of g does not contain x, no additional work is needed as the dependence on x cannot change at g and hence the test in Line 4 is false.

    Otherwise, both reduced children of g contain x. In that case we decide $x \notin \mathrm{var}(g)$ by taking the partial derivative of each reduced child with respect to the variable x and then applying the assumed identity test on Σ^2-read-k formulae to test whether $\frac{\partial g}{\partial x} \not\equiv 0$. Note that $\frac{\partial g}{\partial x}$ is indeed a Σ^2-read-k formula as property (i) implies that the reduced children are read-k formulae. The partial derivative can be computed in time $\mathrm{poly}(s)$ since the multiplication gates in the multilinear subformulae of the reduced children are variable disjoint by property (ii).

Since F is a read-$(k+1)$ formula, for any given variable x, the nontrivial case in 2b) can happen at most $k+1$ times. Moreover, for that case to happen x has to be one of the variables of the original formula F. This implies that $\textsc{Reduce}_k(F)$ makes at most $(k+1)n$ calls to the Σ^2-read-k identity test. All the other work to evaluate Line 4 is $\mathrm{poly}(s)$. Combining the cost for evaluating Line 4 with a straight-forward implementation of the rest of $\textsc{Reduce}_k$, we conclude that our identity test runs in time claimed. □

Since testing multilinear sparse-substituted read-k formulae is trivial for $k = 0$, the special case of Lemma 4.1 with $k = 0$ yields the following corollary.

Corollary 4.3. *There is a deterministic algorithm for identity testing multilinear sparse-substituted read-once formulae that runs in time* $\mathrm{poly}(s)$, *where* s *denotes the size of the formula.*

In fact, the proof of Lemma 4.1 for $k = 0$ shows that Corollary 4.3 even holds without the multilinearity condition (which is non-vacuous for sparse-substituted formulae), but we will not need that extension.

4.2. Blackbox Reduction. Let F be a structurally-multilinear read-$(k+1)$ formula. We construct a generator for F using a generator $\mathcal{G}$ for structurally-multilinear Σ^2-read-k formulae. If F is a read-k formula, the assumed generator alone suffices. Otherwise, we apply the Fragmentation Lemma for structurally-multilinear sparse-substituted formulae (Lemma 3.6) to show that there is a partial derivative of F that has mostly small factors and, possibly, one factor that is a large structurally-multilinear Σ^2-read-k formula. In the former case the factors are small enough to be hit recursively, and in the latter case the factor is hit by the assumed generator for structurally-multilinear Σ^2-read-k formulae. The properties of the SV-generator (Proposition 2.12 and Lemma 2.14) imply that if $\mathcal{G}_n$ is a generator for the partial derivative of a polynomial with n variables, then $\mathcal{G}_n + G_{n,1}$ is a generator for the original polynomial.

Lemma 4.4 (Read-$(k+1)$ PIT $\leq$ Σ^2-Read-k PIT – Blackbox Structurally-Multilinear). *For an integer $k \geq 1$, let $\mathcal{G}$ be a generator for n-variate structurally-multilinear sparse-substituted Σ^2-read-k formulae, and let F be a non-zero n-variable structurally-multilinear sparse-substituted read-$(k+1)$ formula. Then $\mathcal{G} + G_{n,\log|\mathrm{var}(F)|}$ hits F.*

Proof. First observe that if F is read-k, we are immediately done because $F(\mathcal{G}) \not\equiv 0$ and $\bar{0}$ is in the image of the SV-generator (by Proposition 2.12, Part (i)).

The proof goes by induction on $|\mathrm{var}(F)|$. If $|\mathrm{var}(F)| = 0$, the lemma holds trivially as F is constant. If $|\mathrm{var}(F)| = 1$, F is a read-once formula, which is covered by the above observation. For the induction step, by the above observation we can assume that F is read-$(k+1)$ and not read-k. Therefore, F meets the conditions to apply the structurally-multilinear version of the Fragmentation Lemma (Lemma 3.6) with $V = \mathrm{var}(F)$. The lemma produces a variable $x \in \mathrm{var}(F)$ and $\alpha \in \bar{\mathbb{F}}$. The factors of $\partial_{x,\alpha}F$ all depend on at most $\frac{|\mathrm{var}(F)|}{2}$ variables and are read-$(k+1)$ formulae, except for at most one which is a Σ^2-read-k formula. The induction hypothesis gives that the former factors of $\partial_{x,\alpha}F$ are all hit by $\mathcal{G} + G_{n,\log(|\mathrm{var}(F)|/2)}$. The latter factor (if it occurs) is hit by $\mathcal{G}$. Applying Lemma 2.14 gives that $\mathcal{G} + G_{n,\log(|\mathrm{var}(F)|/2)} + G_{n,1}$ hits F. Recalling Proposition 2.12, Part (iii), implies that $\mathcal{G} + G_{n,\log|\mathrm{var}(F)|}$ hits F. □

5. Reducing Testing Σ^2-Read-k Formulae to Testing Read-k Formulae

In this section we present two methods of reducing identity testing structurally-multilinear sparse-substituted Σ^2-read-k formulae to identity testing structurally-multilinear sparse-substituted read-k formulae. We first develop those methods for *multilinear* rather than *structurally-multilinear* sparse-substituted formulae, and then show how to lift them to the latter setting.

Both reductions rely on a common theorem (Theorem 5.4), which we prove in Section 5.2. Informally, that theorem says that for a non-zero multilinear sparse-substituted Σ^2-read-k formula F on n variables and a shift $\bar{\sigma}$ satisfying some simple conditions, the shifted formula $F(\bar{x} + \bar{\sigma})$ is hit by the SV-generator $G_{n,w}$ with $w = k^{O(k)}(\log(t) + 1)$, where t denotes the maximum number of terms in each substituted polynomial.

Note that, since F is a non-zero polynomial, such a theorem is trivially true for a typical shift $\bar{\sigma}$, even with $w = 0$. The interesting part of the theorem is the simplicity of the conditions on $\bar{\sigma}$ that guarantee the hitting property. In particular, the properties needed of $\bar{\sigma}$ allow such a $\bar{\sigma}$ to be computed efficiently either by an identity test for multilinear sparse-substituted read-k formulae, or as an element in the image of a hitting set generator for such formulae.

In Section 5.1 we argue that small sums of specially shifted multilinear sparse-substituted read-k formulae cannot compute a term of high degree. This is the Key Lemma for multilinear sparse-substituted formulae and is a formalization of Lemma 1.3 from the introduction. Using the Key Lemma and the hitting property of the SV-generator (Lemma 2.17), we prove (as Theorem 5.4 in Section 5.2) that the SV-generator hits small sums of specially shifted multilinear sparse-substituted read-k formulae. In Section 5.3 and Section 5.4 we use Theorem 5.4 to argue reductions from identity testing multilinear sparse-substituted Σ^2-read-k formula to identity testing multilinear sparse-substituted read-k formulae in both the non-blackbox and blackbox settings.

Section 5.4 concludes by extending the blackbox reduction to structurally-multilinear sparse-substituted read-k formulae. To do this we use the transformation $\mathcal{L}$ from Section 2.1.4 (see Definition 2.7) to generalize the Key Lemma to structurally-multilinear sparse-substituted formulae, and then argue how the other ingredients transfer. The corresponding step in the non-blackbox setting is straightforward because there $\mathcal{L}$ can be directly applied to efficiently reduce the problem of testing structurally-multilinear sparse-substituted read-k formulae to testing multilinear sparse-substituted read-k formulae.

5.1. Proving the Key Lemma for Multilinear Formulae. In order to prove the Key Lemma, we first establish a similar lemma for *split* multilinear sparse-substituted formulae, and then apply the Shattering Lemma to lift the result to the bounded-read setting.

Let $F = \sum_{i=1}^{m} F_i$ be a sufficiently split multilinear sparse-substituted formula on n variables. By applying the structural witness for split formulae (Lemma 2.19) we can argue that if none of the F_i's are divisible by any variable then F cannot compute a term of the form $a \cdot M_n$, where a is a non-zero constant and, recall, M_n denotes the monomial $\prod_{i=1}^{n} x_i$. The idea is to consider the formula $F - a \cdot M_n$ and apply the structural witness to it in order to show that it is non-zero. The non-divisibility condition and the natural properties of M_n immediately give simplicity. Minimality essentially comes for free because the argument is existential. The splitting required by the structural witness immediately follows from the splitting of F. Formalizing this idea yields the following lemma.

Lemma 5.1. *Let $F = \sum_{i=1}^{m} F_i$ be a multilinear sparse-substituted $\alpha(m+1)$-split formula on $n \geq 1$ variables, where $\alpha \doteq \frac{1}{R}$ and R is the function given by Lemma 2.19. If no F_i is divisible by any variable, then $F \not\equiv a \cdot \prod_{i=1}^{n} x_i$ for any non-zero constant a.*

Note that for a non-constant formula F on n variables to be $\alpha(m+1)$-split, n needs to be at least $1/\alpha(m+1)$.

Proof. Suppose for the sake of contradiction that $F \equiv a \cdot M_n$ for some non-zero constant a.

If there is some subsum of the branches of F that equals 0, eliminate all those branches. Not all branches of F may be eliminated in this way as this contradicts $a \cdot M_n \not\equiv 0$. Let $0 < m' \leq m$ be the remaining number of branches, and let F' denote the sum of the remaining branches. The formula $F' - a \cdot M_n$ is minimal and has top fanin $m' + 1$.

Now, suppose that there is some non-constant polynomial Q that divides every remaining F_i. Since $F' \equiv a \cdot M_n$, Q also divides M_n. Because Q is non-constant, some variable x divides Q and hence divides each remaining F_i. This contradicts the hypothesized non-divisibility property of the F_i. Therefore $F' - a \cdot M_n$ is simple as a formula with top fanin $m' + 1$.

The previous two paragraphs establish that the $F' - a \cdot M_n$ is simple, minimal, and has top fanin $m' + 1$. Further, for every variable, there is some branch that depends on that variable, because the M_n branch depends on every variable. Observe that the M_n branch is trivially $\alpha(m' + 1)$-split and every other branch is also $\alpha(m' + 1)$-split as $m' \leq m$ and $\alpha \doteq \frac{1}{R}$ is decreasing. The structural witness for split formulae (Lemma 2.19) then implies that $F' - a \cdot M_n \not\equiv 0$, and thus that $F \not\equiv a \cdot M_n$. This contradicts the initial assumption and concludes the proof. □

The property that the branches F_i are not divisible by any variable can be easily established by shifting the formula by a point $\bar{\sigma}$ that is a common non-zero of all the branches F_i. Indeed, if we pick $\bar{\sigma}$ such that $F_i(\bar{\sigma}) \neq 0$ then no variable can divide $F_i(\bar{x} + \bar{\sigma})$. This reasoning is formalized in the following corollary.

Corollary 5.2. *Let $F = \sum_{i=1}^{m} F_i$ be a multilinear sparse-substituted $\alpha(m+1)$-split formula on $n \geq 1$ variables, where $\alpha \doteq \frac{1}{R}$ and R is the function given by Lemma 2.19. If no F_i vanishes at $\bar{\sigma}$, then $F(\bar{x} + \bar{\sigma}) \not\equiv a \cdot \prod_{i=1}^{n} x_i$ for any non-zero constant a.*

Proof. Since the branches of F are $\alpha(m+1)$-split, the branches of $F(\bar{x} + \bar{\sigma})$ are also $\alpha(m + 1)$-split. By assumption, $F_i(\bar{\sigma}) \neq 0$. Therefore, for each branch F_i and variable $x_j \in [n]$, $F_i(\bar{x} + \bar{\sigma})|_{x_j \leftarrow 0} \not\equiv 0$. This implies that no variables divide any $F_i(\bar{x} + \bar{\sigma})$. With this property established, apply Lemma 5.1 on $F(\bar{x} + \bar{\sigma})$ to conclude the proof. □

We now show how to lift Corollary 5.2 from split multilinear sparse-substituted formulae to sums of multilinear sparse-substituted bounded-read formulae. This yields our key lemma – that for such formulae F and a "good" shift $\bar{\sigma}$, $F(\bar{x} + \bar{\sigma})$ is not divisible by a term of large degree. For brevity, in the intuition below we discuss the simpler case of showing the formula is, instead, not identical to a term of large degree.

For the sake of contradiction suppose the opposite, i.e., that $F(\bar{x}+\bar{\sigma}) \equiv a \cdot M_n$ for some non-zero constant a and large n. Shatter F into $F' = \partial_P F|_{Z \leftarrow 0}$ using the Shattering Lemma (Lemma 3.7), and apply the same operations that shatter F to M_n. Observe that zero-substitutions are shifted into substitutions by $\bar{\sigma}$, and that $\partial_P M_n|_{Z \leftarrow (-\bar{\sigma})}$ is a non-zero term of degree $n - |P \cup Z|$ provided that no component of $\bar{\sigma}$ vanishes. After an appropriate substitution for variables outside of the set V from the Shattering Lemma, we obtain that $F'(\bar{x}+\bar{\sigma}) \equiv a' \cdot M_V$ for some non-zero constant a' and $V \subseteq [n]$, where M_V denotes the product of the variables in V.

At this point we would like to apply Corollary 5.2 to derive a contradiction. However, we need to have that $|V| > 0$ and that $\bar{\sigma}$ is a common non-zero of all the branches of F'. The former follows from the bounds in the Shattering Lemma provided n is sufficiently large. To achieve the latter condition we impose a stronger requirement on the shift $\bar{\sigma}$ prior to shattering so that afterward $\bar{\sigma}$ is a common non-zero of the shattered branches. The Shattering Lemma tells us that the factors of the branches of the shattered formula are of the form $\partial_{\tilde{P}} f|_{Z \leftarrow 0}$ where f is a subformula of some F_i and $\tilde{P} \subseteq P$. Therefore, we require that $\bar{\sigma}$ is a common non-zero of all such subformulae that are non-zero. This is what we mean by a "good" shift.

One additional technical detail is that we must apply a substitution to the variables outside of V that preserves the properties of $\bar{\sigma}$ and does not zero M_n. This step is in the same spirit as the argument in the proof of the structural witness for split formulae (Lemma 2.19), namely that a typical assignment suffices.

With these ideas in mind, the key lemma is as follows.

Lemma 5.3 (Key Lemma). *Let $F = c + \sum_{i=1}^{m} F_i$, where each $F_i \in \mathbb{F}[x_1, \ldots, x_n]$ is a non-constant multilinear sparse-substituted read-k_i formula, and c is a constant. If $\bar{\sigma}$ is a common non-zero of the non-zero formulae of the form $\partial_P f|_{Z \leftarrow 0}$ where f is a subformula of some F_i and $|P \cup Z| \leq b \doteq (k-m+1) \cdot 4k \cdot R(k+2) \cdot (\log(t)+1)$, then*

$$F(\bar{x}+\bar{\sigma}) \notin \mathcal{D}_\ell,$$

for $\ell \geq w \doteq (8k \cdot R(k+2))^{k-m+1}(\log(t)+1)$, where $k \doteq \sum_{i=1}^{m} k_i$, t

denotes the maximum number of terms in each substituted polynomial, and R is the function given by Lemma 2.19.

Proof. Assume the contrary, without loss of generality, that $F(\bar{x}+\bar{\sigma}) \equiv Q \cdot M_\ell$ for some non-zero multilinear polynomial Q and $\ell \geq w$. If any variable divides Q, factor that variable out and increase ℓ by one. This way we can assume Q is not divisible by any variables.

We first argue that, without loss of generality, $\mathrm{var}(F_i) \subseteq [\ell]$ for all $i \in [m]$. Suppose that some F_i depends on a variable x_j with $j \notin [\ell]$. Replace F with $F|_{x_j \leftarrow \sigma_j}$, and observe this is equivalent to substituting 0 for x_j in $F(\bar{x}+\bar{\sigma})$. We have $M_\ell|_{x_j \leftarrow 0} = M_\ell$, because M_ℓ does not depend on x_j, and $Q' \doteq Q|_{x_j \leftarrow 0} \not\equiv 0$, because x_j does not divide Q. The assignment $\bar{\sigma}$ remains a common non-zero of the stated type of formulae, now with F_i replaced by $F_i|_{x_j \leftarrow \sigma_j}$. If Q' is divisible by any variables factor them out, and increase ℓ accordingly. Repeat this procedure until $\mathrm{var}(F_i) \subseteq [\ell]$ for all $i \in [m]$.

Note that these substitution may make some branches constant. In this case combine these constant branches into a single constant branch. Since all F_i were originally non-constant, the quantity $k - m$ has not increased.

Define $\alpha \doteq \frac{1}{R}$. Shatter F using Lemma 3.7. This produces the sets of variables P, Z, and V. Let $F' \doteq \partial_P F|_{Z \leftarrow 0}$. By the Shattering Lemma $F' = c' + \sum_{i=1}^{m'} F'_i$ is a multilinear sparse-substituted formula that has top fanin $m'+1 \leq k+1$, is $\alpha(m'+2)$-split$_V$, and each F'_i is a product of factors of formulae of the form $\partial_{\tilde{P}} f|_{Z \leftarrow 0}$ where f is a subformula of an F_i and $\tilde{P} \subseteq P$. Assume without loss of generality that each F'_i is non-zero. By the lemma, $|P \cup Z| \leq b$. By hypothesis, the subformulae of the above form do not vanish at $\bar{\sigma}$. These properties imply that $F'_i(\bar{\sigma}) \neq 0$ for each $i \in [m']$.

There is an assignment to the variables in $[\ell] \backslash V$ that: (1) preserves $\bar{\sigma}$ as a non-zero of the F'_i's on the remaining variables V, and (2) differs in every component from $\bar{\sigma}$. In fact, a typical assignment suffices. To see this, consider the polynomial:

$$\Phi \doteq \left(\prod_{i=1}^{m'} F'_i|_{V \leftarrow \bar{\sigma}} \right) \cdot \prod_{j \in ([\ell] \backslash V)} (x_j - \sigma_j).$$

The polynomial Φ is non-zero because the F_i''s do not vanish at $\bar{\sigma}$. Thus, a non-zero assignment for Φ satisfies the requirements above. Pick $\bar{\beta}$ to be any such assignment.

Let $F'' \doteq F'|_{([\ell]\setminus V)\leftarrow\bar{\beta}}$, where the F_i'' are defined similarly. By the first property of $\bar{\beta}$, $F_i''(\bar{\sigma}) \neq 0$. By the second property of $\bar{\beta}$, $M_\ell|_{([\ell]\setminus V)\leftarrow(\bar{\beta}-\bar{\sigma})}$ is a non-zero term over the variables V. Then using the initial assumption write

$$F''(\bar{x}+\bar{\sigma}) \equiv F'(\bar{x}+\bar{\sigma})|_{([\ell]\setminus V)\leftarrow(\bar{\beta}-\bar{\sigma})} \equiv a \cdot M_\ell|_{([\ell]\setminus V)\leftarrow(\bar{\beta}-\bar{\sigma})} \equiv a' \cdot M_V,$$

for some non-zero constant a'. Now, $F'' \in \mathbb{F}[V]$ is a multilinear sparse-substituted $\alpha(m'+2)$-split$_V$ formula with top fanin $m'+1$, where no branch vanishes at $\bar{\sigma}$. Thus, we obtain a contradiction with Corollary 5.2 as long as $|V| > 0$. By the bound on $|V|$ given in the Shattering Lemma and then condition that $\ell \geq w$, the latter is the case for $w \geq (8k \cdot R(k+2))^{k-m+1}(\log(t)+1)$. □

5.2. Generator for Shifted Multilinear Formulae. In this subsection we show that the SV-generator hits small sums of specially shifted multilinear sparse-substituted bounded-read formulae. Our argument critically relies on the property given in Lemma 2.17 – that the SV-generator hits any class of polynomials that is closed under zero-substitutions and such that no term of high degree divides polynomials in the class.

In order to prove a usable theorem for our applications, we use the Key Lemma (Lemma 5.3) to construct a class of polynomials sufficient to apply Lemma 2.17. Let F be a formula, $\bar{\sigma}$ be a shift, and w be as in the statement of the Key Lemma. Consider $F(\bar{x}+\bar{\sigma})$. By the Key Lemma, $F(\bar{x}+\bar{\sigma}) \notin \mathcal{D}_n$, for $n \geq w$. Now consider substituting 0 for x_j in $F(\bar{x}+\bar{\sigma})$, this equivalent to substituting σ_j for x_j in F then shifting all other variables by $\bar{\sigma}$. This means that the preconditions of the Key Lemma are satisfied for $F|_{x_j\leftarrow\sigma_j}$, and hence $F(\bar{x}+\bar{\sigma})|_{x_j\leftarrow 0} \notin \mathcal{D}_n$, for $n \geq w$. This argument can be repeated to get that each zero-substitution of $F(\bar{x}+\bar{\sigma})$ is not in $\mathcal{D}_n$, for $n \geq w$. The set of polynomials which corresponds to all zero-substitutions of $F(\bar{x}+\bar{\sigma})$ serves as the set $\mathcal{P}$ in the application of Lemma 2.17. This, in turn, implies that $G_{n,w}$ hits $F(\bar{x}+\bar{\sigma})$, since it is a member of this set of polynomials.

Theorem 5.4. *Let $F = c + \sum_{i=1}^{m} F_i$, where c is a constant, and each $F_i \in \mathbb{F}[x_1, \ldots, x_n]$ is a non-constant multilinear sparse-substituted read-k_i formula. If $\bar{\sigma}$ is a common non-zero of the non-zero formulae of the form $\partial_P f|_{Z\leftarrow 0}$ where f is a subformula of some F_i and $|P \cup Z| \leq b \doteq (k - m + 1) \cdot 4k \cdot R(k+2) \cdot (\log(t) + 1)$, then*

$$F \not\equiv 0 \Rightarrow F(G_{n,w} + \bar{\sigma}) \not\equiv 0$$

for $w \geq (8k \cdot R(k+2))^{k-m+1}(\log(t) + 1)$, where $k \doteq \sum_{i=1}^{m} k_i$, t denotes the maximum number of terms in each substituted polynomial, and R is the function given by Lemma 2.19.

Proof. Define the classes of formulae

$$\mathcal{F} \doteq \{F|_{S\leftarrow\bar{\sigma}} \mid S \subseteq [n]\}, \text{ and } \mathcal{F}' \doteq \{F'(\bar{x} + \bar{\sigma}) \mid F' \in \mathcal{F}\}.$$

Observe that $\mathcal{F}'$ is closed under zero-substitutions because $\mathcal{F}$ is closed under substitutions by $\bar{\sigma}$ and for any variable x_j, $F'(\bar{x} + \bar{\sigma})|_{x_j\leftarrow 0} = (F'|_{x_j\leftarrow\sigma_j})(\bar{x} + \bar{\sigma})$.

Without loss of generality each $F' \in \mathcal{F}$ has at most one top level branch which is constant, since constant branches can be collected into a single constant branch without compromising any of the relevant properties of F'. Observe that for each $F' \in \mathcal{F}$, the assignment $\bar{\sigma}$ remains a common non-zero of the subformulae of F' under at least b partial derivatives and zero-substitutions, because we are performing a partial substitution of $\bar{\sigma}$ itself. Therefore, for each $F' \in \mathcal{F}$, the preconditions of Lemma 5.3 are met and hence $F'(\bar{x}+\bar{\sigma}) \notin \mathcal{D}_\ell$, for $\ell \geq w$. This implies that $\mathcal{F}'$ is disjoint from $\mathcal{D}_\ell$, for $\ell \geq w$. Lemma 2.17 then says that $G_{n,w}$ hits $\mathcal{F}'$, and $F(\bar{x} + \bar{\sigma})$ in particular. □

5.3. Non-Blackbox Reduction. In this subsection we focus on giving a non-blackbox reduction from identity testing multilinear sparse-substituted Σ^m-read-k formulae to identity testing multilinear sparse-substituted read-k formulae on n variables. The extension from multilinear to structurally-multilinear formulae follows immediately by the transformation $\mathcal{L}$ from Section 2.1.4, and we incorporate it into the proof of the main non-blackbox result in Section 6.1.

The first step of the reduction is to compute an appropriate shift $\bar{\sigma}$ using an identity test for multilinear sparse-substituted read-k formulae. One technical complication is to ensure that the formula has gates that are explicitly multilinear, so that partial derivatives can be computed efficiently. This can be done using an identity test for multilinear sparse-substituted read-k formulae. We also evaluate the constant parts of the formula in an effort to reduce the effective number of subformulae that must be tested recursively when computing $\bar{\sigma}$. Once we have $\bar{\sigma}$, we simply evaluate $F(G_{n,w}+\bar{\sigma})$ on sufficiently many points and see whether we obtain a nonzero value.

Lemma 5.5 (Σ^m-Read-k PIT $\leq$ Read-k PIT – Non-Blackbox Multilinear). *For any integer $k \geq 1$, given a deterministic identity test for multilinear sparse-substituted read-k formulae that runs in time $T(k,n,s,t)$, there is a deterministic identity test for multilinear sparse-substituted Σ^m-read-k formulae that runs in time*

$$k^2m^2n^{O(b)} \cdot T(k, n, O(s\log(kmn^{b+3}))), t) + n^{O(w_{m,k}\cdot(\log(t)+1))}\,\mathrm{poly}(s),$$

where s denotes the size of the formula, n the number of variables, and t the maximum number of terms in each substituted polynomial, $b \doteq ((k-1)m+1)\cdot 4km \cdot R(km+2)\cdot(\log(t)+1)$, $w_{m,k} \doteq (8km \cdot R(km+2))^{(k-1)m+1}$, and R is the function given by Lemma 2.19.

Proof. Let $F \doteq \sum_{i=1}^{m} F_i$, where each F_i is a multilinear sparse-substituted read-k formula. Let b be sufficient to apply Theorem 5.4 with the parameters $k_i = k$, m, and n.

Ensuring syntactic multilinearity. Observe that since F_i is a read-k formula there are at most kn gates in F_i whose children both contain variable occurrences. Call these gates the *essential* gates of F_i. Process each of the F_i from the bottom up, making the children of multiplication gates variable disjoint. To do this, at each essential gate g in F_i compute the set of variables that g depends on. This can be done using the hypothesized identity test on the first order partial derivatives of g with respect to each variable. These partial derivatives can be efficiently computed as the children have been previously processed to have variable disjoint multiplication gates. Set variables that g does not depend on to 0, though only within the subformula g. Note that this does not affect the polynomial computed at each gate of F_i; it merely removes extraneous variable occurrences. Since F_i has at most kn essential gates, this step uses at most kmn^2 applications of the read-k identity test and a poly(s) amount of local computation.

Shrinking the number of subformulae. Let g be an essential gate of F_i. Let g' be the unique gate above g which is either the output or a child of an essential gate (it may be that $g' = g$). Write $g' = \alpha_g g + \beta_g$ for constants α_g and β_g determined by evaluating the constant part of the formula between g and g'.

Let $\mathcal{F}$ be the set of all variables $\{x_j\}_{j \in [n]}$ unioned with the set of all non-zero formulae of the form $\partial_P f|_{Z \leftarrow 0}$ where f is one of $\{g, \alpha_g g, \alpha_g g + \beta_g\}$ for some essential gate g of a branch F_i, and $|P \cup Z| \leq b$. Notice that the elements of $\mathcal{F}$ are multilinear sparse-substituted read-k formulae because each g is of that type, and that type of formulae is closed under partial derivatives and substitutions. The elements of $\mathcal{F}$ have size s as multilinear sparse-substituted read-k formulae. To see this, observe that g is explicitly a subformula of F_i and hence has size s. Let g' be the unique gate above g which is either the output of F_i or a child of an essential gate. By construction $g' \equiv \alpha_g g + \beta_g$ and since g' is a subformula of F_i, the polynomial $\alpha_g g + \beta_g$ can be expressed by a size s multilinear sparse-substituted read-k formula. Finally, note that the polynomial $\alpha_g g$ can be expressed as a size s formula by taking g' and dropping all addition branches that diverge from the path between g' and g (i.e., $\alpha_g = \partial_g g'$). There are at most $n + 3kmn^{b+1} \leq 4kmn^{b+1}$ formulae in $\mathcal{F}$. Since the multiplication gates of F are variable disjoint this implies that the formulae in $\mathcal{F}$ can be enumerated in time $|\mathcal{F}| \cdot \text{poly}(s)$.

Finding a common non-zero of $\mathcal{F}$. Define the polynomial $\Phi \doteq \prod_{f \in \mathcal{F}} f \not\equiv 0$. All the formulae in $\mathcal{F}$ are multilinear. This means that Φ has total degree at most $4kmn^{b+2}$. By the Schwartz-Zippel Lemma we only need to test elements from a subset W from $\mathbb{F}$ (or an extension field of $\mathbb{F}$) of size at most the degree of Φ plus one. We can use trial substitution to determine a point, $\bar{\sigma} \in \mathbb{F}^n$

where Φ is non-zero, in a manner similar to finding a satisfying assignment to a CNF formula given a SAT oracle: For each variable, in turn, determine a value from W that keeps Φ non-zero. Fix the variable to this value, and then move on to consider the next variable. This uses at most $8kmn^{b+3}$ identity tests on a partially substituted version of Φ. Each of these identity tests on Φ uses at most $4kmn^{b+1}$ identity tests to test the individual factors of Φ. In total, our algorithm uses at most $72k^2m^2n^{2b+4}$ identity tests on multilinear sparse-substituted read-k formulae to compute $\bar{\sigma}$. Note that the substitution of values from W may increase the individual size of the formulae in $\mathcal{F}$ by an additive amount of $O(s\log(kmn^{b+3}))$ overall. Thus we can compute $\bar{\sigma}$ in $72k^2m^2n^{2b+4} \cdot T(k, n, s(1 + O(\log(kmn^{b+3}))), t)$ time using the assumed identity test.

Putting everything together. We conclude by arguing that $\bar{\sigma}$ suffices to apply Theorem 5.4 to show that $G_{n,w_{m,k}\cdot(\log t+1)} + \bar{\sigma}$ hits F. Suppose we recursively transform F, by replacing the subformula g' rooted at the root or at a child of an essential gate in a branch F_i, with the formula $\alpha_g g + \beta_g$ where g is the first essential descendant of g'. The resulting formula $\hat{F}$ is equivalent to F, but has at most $3kmn$ gates. Moreover for every subformula $\hat{f}$ of $\hat{F}$ and sets $|P \cup Z| \leq b$, the polynomial $\partial_P \hat{f}|_{Z\leftarrow 0}$ (if non-zero) appears in $\mathcal{F}$ by construction. This implies that $\bar{\sigma}$ is satisfies the conditions for Theorem 5.4 with respect to $\hat{F}$. Since $\hat{F} \equiv F$, $F \equiv 0$ iff $F(G_{n,w_{m,k}\cdot(\log t+1)} + \bar{\sigma}) \equiv 0$. By multilinearity, the formula F has degree at most n and the SV-generator has degree n. Since the SV-generator is computable in polynomial time, applying Proposition 2.10 gives a test for $F(\bar{x} + \bar{\sigma})$ that runs in time $n^{O(w_{m,k}\cdot(\log t+1))}\,\mathrm{poly}(s)$. This is the identity test we desired. Combining the running time for all parts gives the total running time claimed. □

5.4. Blackbox Reduction. We begin by describing a blackbox version of Lemma 5.5, i.e., a blackbox reduction for multilinear sparse-substituted formulae. The overall approach is the same, though the details are somewhat simpler. With Theorem 5.4 in hand, all that remains is to demonstrate an appropriate shift lies in the image of a generator for multilinear sparse-substituted read-k formulae and then apply the theorem to complete the reduction.

LEMMA 5.6 (Σ^m-Read-k PIT $\leq$ Read-k PIT – Blackbox Multilinear). *For an integer $k \geq 1$, let $\mathcal{G}$ be a generator for n-variate multilinear sparse-substituted read-k formulae. Then $\mathcal{G}+G_{n,w_{m,k}\cdot(\log(t)+1)}$ is a generator for n-variate multilinear sparse-substituted Σ^m-read-k formulae, where $w_{m,k} \doteq (8km \cdot R(km+2))^{(k-1)m+1}$, t denotes the maximum number of terms in each substituted polynomial, and R is the function given by Lemma 2.19.*

PROOF. Let F be a multilinear sparse-substituted Σ^m-read-k formula. Write $F \doteq \sum_{i=1}^{m} F_i$, where each F_i is a multilinear sparse-substituted read-k formula.
Let $b \doteq ((k-1)m+1) \cdot 4km \cdot R(km+2) \cdot (\log(t)+1)$ and $w \doteq w_{m,k} \cdot (\log(t)+1)$; in other words, sufficient parameters for applying Theorem 5.4 with m and $k_i = k$.

Let $\mathcal{F}$ be the set of all non-zero formulae of the form $\partial_P f|_{Z\leftarrow 0}$ where f is a subformula of some F_i and P, Z are disjoint sets of variables with $|P \cup Z| \leq b$. Consider the polynomial $\Phi \doteq \prod_{f\in\mathcal{F}} f$. Note that $\Phi \not\equiv 0$, and that each $f \in \mathcal{F}$ is multilinear sparse-substituted read-k formula with at most t terms in each substituted polynomial.

Since $\mathcal{G}$ is a generator for multilinear sparse-substituted read-k formulae and Φ is the product of multilinear sparse-substituted read-k formula, $\mathcal{G}$ hits Φ. Consequently, there is a point $\bar{\beta}$ with components in a finite extension $\mathbb{E} \supseteq \mathbb{F}$ that witnesses the non-zeroness of $\Phi(\mathcal{G})$. This implies that no formula in $\mathcal{F}$ vanishes at $\mathcal{G}(\bar{\beta})$. By Theorem 5.4, $G_{n,w_{m,k}\cdot(\log(t)+1)}$ hits $F(\bar{x}+\mathcal{G}(\bar{\beta}))$. Thus, $\mathcal{G}+G_{n,w_{m,k}\cdot(\log(t)+1)}$ hits F, completing the reduction. □

In the blackbox setting the extension from multilinear sparse-substituted to structurally-multilinear sparse-substituted formulae takes some work. We first need to extend the Key Lemma (Lemma 5.3).

5.4.1. Generalizing the Key Lemma. The statement of the generalization for structurally-multilinear sparse-substituted formulae is almost identical to the original one for multilinear sparse-substituted formulae, except that $\bar{\sigma}$ must be the common non-zero of more formulae. The proof is via a reduction to the original lemma. Here is the outline.

Let F be a structurally-multilinear formula. Suppose that $F(\bar{x}+\bar{\sigma}) \in \mathcal{D}_\ell$ for some assignment $\bar{\sigma}$. Recall the transformation $\mathcal{L}$ discussed in Section 2.1.4. By a hybrid argument we show that for each variable x_j there is a degree d_j such that substituting the appropriate power of σ_j into the variables $y_{j,d}$, for $d < d_j$, makes the multilinear sparse-substituted formula $\mathcal{L}(F)$ divisible by the linear polynomial $(y_{j,d_j} - \sigma_j^{d_j})$. Doing this to $\mathcal{L}(F)$ for each $j \in [n]$ produces a formula which is divisible by a shifted monomial in the y_{j,d_j} variables. The only variables $y_{j,d}$ that remain have $d = d_j$, or $d > d_j$. The variables y_{j,d_j} will be the "x" variables when we apply the Key Lemma. We fix the variables $y_{j,d}$, for $d > d_j$, to a typical substitution, so that the relevant properties are preserved. The result is a multilinear sparse-substituted formula in the variables y_{j,d_j} which computes a shifted monomial. This allows us to reach a contradiction by applying Lemma 5.3.

The remaining question is: What are the conditions on $\bar{\sigma}$? $\bar{\sigma}$ must be a common non-zero of all the non-zero subformulae that may be considered by the Key Lemma when the above process is complete. However, we do not know *a priori* which choices our proof makes for the d_j, and hence which variables remain when applying the Key Lemma. Therefore, we require that $\bar{\sigma}$ be a common non-zero with respect to all possible choices of the d_j. In particular, we want $\bar{\sigma}$ to be the common non-zero of the non-zero $\partial_P(\mathcal{L}_{X(P\cup Z)}(f))|_{Z\leftarrow 0}$ where f is a subformula of F, and P and Z are sets of $y_{j,d}$ variables. This way, independent of the choices the proof makes for the d_j the conditions of the Key Lemma can be satisfied.

This intuition is formalized the following lemma.

Lemma 5.7 (Generalized Key Lemma). *Let $F = c + \sum_{i=1}^{m} F_i$, where c is a constant, and each $F_i \in \mathbb{F}[x_1,\ldots,x_n]$ is a non-constant structurally-multilinear sparse-substituted read-k_i formula. If $\bar{\sigma}$ is a common non-zero of the non-zero formulae of the form $\partial_P(\mathcal{L}_{X(P\cup Z)}(f))|_{Z\leftarrow 0}$ where f is a subformula of some F_i and $P, Z \subseteq$*

$Y \doteq \{y_{\ell,j} \mid \ell, j \geq 1\}$ *such that*

$$|P \cup Z| \leq b \doteq (k - m + 1) \cdot 4k \cdot R(k + 2) \cdot (\log(t) + 1),$$

then

$$F(\bar{x} + \bar{\sigma}) \notin \mathcal{D}_\ell,$$

for $\ell \geq w \doteq (8k \cdot R(k+2))^{k-m+1}(\log(t) + 1)$, *where* $k \doteq \sum_{i=1}^{m} k_i$, t *denotes the maximum number of terms in each substituted polynomial, and* R *is the function given by Lemma 2.19.*

Proof. Assume the contrary, without loss of generality, that $F(\bar{x} + \bar{\sigma}) \equiv Q \cdot M_\ell$ for some non-zero polynomial Q and $\ell \geq w$. Denote $\hat{F} = \sum_{i=1}^{m} \hat{F}_i \doteq \mathcal{L}(F)$. As $\hat{F} \not\equiv 0$, for each $j \in [n]$ there must exist maximum $d_j \geq 1$ such that

$$\hat{F}|_{\left\{y_{j,d} \leftarrow \sigma_j^d \ \mid\ d \in [d_j - 1]\right\}} \not\equiv 0$$

Observe that for each $j \in [\ell]$, $F|_{x_j \leftarrow \sigma_j} \equiv 0$. In addition, by Lemma 2.8, Part (ii), and the definition of $\mathcal{L}$:

$$\begin{aligned} 0 &\equiv \mathcal{L}(F|_{x_j \leftarrow \sigma_j}) \\ &= \mathcal{L}(F)|_{\left\{y_{j,d} \leftarrow \sigma_j^d \ \mid\ d \geq 1\right\}} \\ &= \hat{F}|_{\left\{y_{j,d} \leftarrow \sigma_j^d \ \mid\ d \geq 1\right\}} \end{aligned}$$

This means that $\hat{F}|_{\left\{y_{j,d} \leftarrow \sigma_j^d \ \mid\ d \in [d_j - 1]\right\}}$ has $(y_{j,d_j} - \sigma_j^{d_j})$ as a factor. By repeating this argument sequentially for every $j \in [n]$, and using the fact that substitutions on multilinear polynomials commute, we obtain a sequence $(d_1, \ldots, d_n) \in \mathbb{N}^n$ such that

$$\hat{F}' = \sum_{i=1}^{m} \hat{F}'_i \doteq \hat{F}|_{\left\{y_{j,d} \leftarrow \sigma_j^d \ \mid\ j \geq 1\ ,\ d \in [d_j - 1]\right\}} \not\equiv 0.$$

Moreover, $\hat{F}'$ is a multilinear sparse-substituted m-sum of read-k_i formulae and

$$\hat{F}' \equiv Q' \cdot \prod_{j \in [\ell']} (y_{j,d_j} - \sigma_j^{d_j})$$

for some non-zero polynomial Q' and integer ℓ'. We have $\ell' \geq \ell$ because each $j \in [\ell]$ must produce such a linear factor (the $j \in [n]\backslash[\ell]$ may or may not contribute such factors). Partition $Y = \{y_{j,d} \mid j, d \geq 1\}$ into three sets depending on whether $d < d_j$, $d = d_j$, or $d > d_j$. Call these three sets $Y^<$, $Y^=$, and $Y^>$ respectively.

Consider a subformula $\hat{f}'$ of some $\hat{F}_i'$ and let f and $\hat{f}$, respectively, be the corresponding subformulae of F_i and $\hat{F}_i$. Let $P, Z \subseteq Y^=$ be such that $|P \cup Z| \leq b$ and $\partial_P \hat{f}'|_{Z\leftarrow 0} \not\equiv 0$. By Lemma 2.8, Part (iv), and then the definition of $\bar{\sigma}$:

$$\begin{aligned}&\left(\partial_P \hat{f}|_{Z\leftarrow 0}\right)\Big|_{\left\{y_{j,d}\leftarrow \sigma_j^d \,\mid\, j,d\geq 1\right\}} \\ &\quad = \left(\partial_P(\mathcal{L}_{X(P\cup Z)}(f))|_{Z\leftarrow 0}\right)\Big|_{\{x_j\leftarrow \sigma_j \,\mid\, j\geq 1\}} \not\equiv 0.\end{aligned}$$

The substitution on the LHS of the above equation can be partitioned corresponding to the sets $Y^<, Y^=$, and $Y^>$. We drop the substitutions associated with $Y^>$; this keeps the formula non-zero. Since $\hat{f}$ is multilinear, $P, Z \subseteq Y^=$, and $Y^=$ is disjoint from $Y^<$, the substitutions of variables from $Y^<$ commutes with partial derivatives on P and zero-substitutions on Z. This fact allows us to push the substitutions over $Y^<$ closer to $\hat{f}$ (to form $\hat{f}'$), and reach the following conclusion

$$\begin{aligned}&\left(\partial_P \hat{f}'|_{Z\leftarrow 0}\right)\Bigg|_{\left\{y_{j,d_j}\leftarrow \sigma_j^{d_j} \,\mid\, j\geq 1\right\}} \\ &\quad \equiv \left(\partial_P \hat{f}|_{Z\leftarrow 0}\right)\Big|_{\left\{y_{j,d}\leftarrow \sigma_j^d \,\mid\, j\geq 1, d\in[d_j]\right\}} \not\equiv 0.\end{aligned}$$

This argument shows that substituting $\sigma_j' \doteq \sigma_j^{d_j}$ for all $j \in [n]$ into $Y^=$ does not zero the formula $\partial_P \hat{f}'|_{Z\leftarrow 0}$. Moreover, this argument is generic with respect to the choice of $\hat{f}'$, P, and Z, so the substitution $\bar{\sigma}'$ for $Y^=$ does not zero $\partial_P \hat{f}'|_{Z\leftarrow 0}$ for any subformula $\hat{f}'$ of $\hat{F}'$, and any choice of disjoint $P, Z \subset Y^=$ satisfying $|P \cup Z| \leq b$.

However, the resulting formulae are over $Y^> \cup Y^=$ not just $Y^=$. Observe that Q' may only depend on variables from $Y^>$ because $\hat{F}'$ is multilinear. Fix the variables of $Y^>$ so that $\bar{\sigma}'$ is a common non-zero of the formulae $\partial_P \hat{f}'|_{Z\leftarrow 0}$, and Q' is not zeroed (for this, a typical substitution suffices). Let $\hat{F}''$ be the result of applying this substitution to $\hat{F}'$. We have that

$$\hat{F}'' = Q'' \cdot \prod_{j \in [\ell']} (y_{j,d_j} - \sigma'_j),$$

for some non-zero polynomial Q'', is a multilinear sparse-substituted formula over only the variables in $Y^=$. Define $x'_j \doteq y_{j,d_j}$ for all $j \in [n]$. Then $\hat{F}''(\bar{x}' + \bar{\sigma}')$ is a term of degree ℓ'. Furthermore, we argued that for all subformulae $\hat{f}''$ of $\hat{F}''$ and disjoint sets $P, Z \subset Y^=$, with $|P \cup Z| \leq b$, we have that $\bar{\sigma}'$ does not zero $\partial_P \hat{f}''|_{Z \leftarrow 0}$. Collect the constant branches of $\hat{F}''$ into a single constant branch; the resulting formula satisfies all preconditions of Lemma 5.3, and a contradiction immediately follows. □

Note that if the given structurally-multilinear formula is in fact a multilinear formula, then the conditions of the lemma are equivalent to the conditions of Lemma 5.3 (up to a relabeling of the variables). Also note the proof of Lemma 5.7 only used sets P and Z that are disjoint, and that contain at most one y variable that corresponds to each x_j, so we could have relaxed the statement of the lemma accordingly.

5.4.2. Blackbox Reduction for Structurally-Multilinear Formulae. We are now ready to generalize Lemma 5.6 to structurally-multilinear formulae. We first observe that Theorem 5.4 generalizes to structurally-multilinear formulae.

Theorem 5.8. *Let $F = c + \sum_{i=1}^{m} F_i$, where c is a constant, and each F_i is a non-constant structurally-multilinear sparse-substituted read-k_i formula on n variables. If $\bar{\sigma}$ is a common non-zero of the non-zero formulae of the form $\partial_P(\mathcal{L}_{X(P \cup Z)}(f))|_{Z \leftarrow 0}$ where f is a subformula of some F_i and $P, Z \subseteq \{y_{\ell,j} \mid \ell, j \geq 1\}$, $|P \cup Z| \leq b \doteq (k - m + 1) \cdot 4k \cdot R(k+2) \cdot (\log(t) + 1)$, then*

$$F \not\equiv 0 \Rightarrow F(G_{n,w} + \bar{\sigma}) \not\equiv 0,$$

for $w \geq (8k \cdot R(k+2))^{k-m+1}(\log(t) + 1)$, where $k \doteq \sum_{i=1}^{m} k_i$, t denotes the maximum number of terms in each substituted polynomial, and R is the function given by Lemma 2.19.

Proof. Observe that the statement of this theorem is the same as Theorem 5.4 except that it takes on the conditions associated with Generalized Key Lemma (Lemma 5.7). To prove this theorem follow the proof of Theorem 5.4, but use the stronger preconditions to apply Lemma 5.7 instead of Lemma 5.3. □

With Theorem 5.8 in hand, we can argue the following generalization of Lemma 5.6. The statement is identical to the original except that "multilinear sparse-substituted" is replaced by "structurally-multilinear sparse-substituted".

Lemma 5.9 (Σ^m-Read-k PIT $\leq$ Read-k PIT – Blackbox Structurally-Multilinear). *For an integer $k \geq 1$, let $\mathcal{G}$ be a generator for n-variate structurally-multilinear sparse-substituted read-k formulae. Then $\mathcal{G} + G_{n,w_{m,k}\cdot(\log(t)+1)}$ is a generator for n-variate structurally-multilinear sparse-substituted Σ^m-read-k formulae, where $w_{m,k} \doteq (8km \cdot R(km+2))^{(k-1)m+1}$, t denotes the maximum number of terms in each substituted polynomial, and R is the function given by Lemma 2.19.*

Proof. The proof is the same as in the original version except that Theorem 5.8 is applied instead of Theorem 5.4. This means that the class $\mathcal{F}$ of polynomials which $\bar{\sigma}$ is a common non-zero of must be larger to account for the stronger preconditions of the theorem. □

6. Identity Testing Read-k Formulae

Before moving on to prove our main theorems, we briefly stop to recall the overall approach. For clarity we only state the non-blackbox approach; the blackbox approach follows a similar pattern. We construct an identity test for structurally-multilinear read-k formulae using four tools.

Lemma 4.1 – a reduction from identity testing multilinear sparse-substituted read-$(k+1)$ formulae to identity testing multilinear sparse-substituted Σ^2-read-k formulae.

Lemma 5.5 – a reduction from identity testing multilinear sparse-substituted Σ^2-read-k formulae to identity testing multilinear sparse-substituted read-k formulae.

Corollary 4.3 – an identity test for multilinear sparse-substituted read-once formulae.

Lemma 2.8 (Parts (i) & (ii)) – a reduction from identity testing structurally-multilinear read-k formulae to identity testing multilinear sparse-substituted read-k formulae.

Combining the first two reductions reduces identity testing multilinear sparse-substituted read-$(k+1)$ formulae to identity testing multilinear sparse-substituted read-k formulae. Applying this observation recursively and combining it with Corollary 4.3 as the base case, establishes an identity test for multilinear sparse-substituted read-k for arbitrary (not necessarily constant) k. We then plug in Lemma 2.8 to lift this result to structurally-multilinear formulae.

In the blackbox setting we deal directly with structurally-multilinear formulae. In the last subsection we develop a specialized blackbox identity test for structurally-multilinear sparse-substituted read-k formulae of constant depth.

6.1. Non-Blackbox Identity Test. Combining Lemmas 4.1, 5.5, 2.8, and Corollary 4.3 in the way suggested above proves the following main result.

Theorem 6.1 (Main Result – Non-Blackbox). *There exists a deterministic polynomial identity test for structurally-multilinear sparse-substituted formulae that runs in time*

$$s^{O(1)} \cdot (dn)^{k^{O(k)}(\log(t)+1)},$$

where s denotes the size of the formula, n the number of variables, k the maximum number of substitutions in which a variable appears, t the maximum number of terms a substitution consists of, and d the maximum degree of individual variables in the substitutions.

Proof. Consider a structurally-multilinear sparse-substituted read-k formula F. In time polynomial in the size of F we can compute the transformed formula $F' \doteq \mathcal{L}(F)$ given in Definition 2.7. By Lemma 2.8, Parts (i) and (ii), F is non-zero iff F' is, and F' is a multilinear sparse-substituted read-k formula of size $s' = \mathrm{poly}(s)$ on $n' \leq dn$ variables where each substitution contains at most $t' = t$ terms.

In order to identity test F', we apply a recursive procedure based on the reductions from Lemma 4.1 and Lemma 5.5, with the base case given by Corollary 4.3. It remains to analyze the running time $T(k, n', s', t')$ of the resulting algorithm, which we do by induction on k.

By Corollary 4.3, $T(1, n', s', t') = \mathrm{poly}(s')$. Consider the induction step from k to $k + 1$. By Lemma 5.5 we can test multilinear sparse-substituted Σ^2-read-k formulae of size s' on n' variables with sparsity t' in time $k^2 n'^b \cdot T(k, n', s'b \log(2kn'), t') + n'^w \cdot \mathrm{poly}(s'))$, where $b = O(k^4 \log k \cdot (\log(t') + 1))$ and $w = k^{O(k)}(\log(t') + 1)$. By Lemma 4.1 this means that $T(k + 1, n', s', t') = O(kn'(k^2 n'^b \cdot T(k, n', s'b \log(2kn'), t') + n'^w \cdot \mathrm{poly}(s')) + \mathrm{poly}(s'))$. Moreover, the proofs of Lemma 4.1 and Lemma 5.5 show that the reductions can be uniformly constructed from k and that the constants hidden in the Big-Oh notations are independent of k. Solving the recurrence for T and using the facts that $n' \leq dn$, $s' = \mathrm{poly}(s)$, and $t' = t$, we obtain the claimed bound. □

This theorem instantiates to the non-blackbox part of Theorem 1.2 when the read k is constant and further to the non-blackbox part of Theorem 1.1 when $t = d = 1$. Using transformations different from $\mathcal{L}$ (Definition 2.7) it is possible to attain alternate (often incomparable) running-time parameterizations in the main theorem.

6.2. Blackbox Identity Test. We proceed analogously to the previous subsection but skip the intermediate step of multilinear sparse-substituted formulae. We first argue that the SV-generator works for structurally-multilinear sparse-substituted read-once formulae – this extends the argument in Shpilka & Volkovich (2009), which works for read-once formulae. Additionally, the argument is stated with respect to a depth parameter to make a later specialization to constant-depth more concise.

The idea is the following. We recurse on the structure of the structurally-multilinear sparse-substituted read-once formula F and argue that the SV-generator takes non-constant subformulae to non-constant subformulae. There are three generic cases, based on the top gate of F: (i) addition, (ii) multiplication, and (iii) a sparse-substituted input.

In case (i), the fact that F is read-once implies that addition branches are variable disjoint. This means that there is a variable whose partial derivative eliminates at least half of the formula and reduces the depth by one. Combining this fact with Lemma 2.14 completes the case. In case (ii), the fact that the SV-generator takes non-constant subformulae to non-constant subformulae immediately implies that if the SV-generator hits the children of a multiplication gate it also hits the gate itself. In case (iii) we can immediately conclude using Lemma 2.15.

Lemma 6.2. *Let F be a non-zero structurally-multilinear sparse-substituted depth-D read-once formula on n variables. Then $G_{n,w}$ hits F for $w \doteq \min\{\lceil \log|\mathrm{var}(F)|\rceil, D\} + \lceil \log t \rceil + 1$, where t denotes the maximum number of terms in each substituted polynomial. Moreover, if F is non-constant then so is $F(G_{n,w})$.*

Proof. We proceed by structural induction on F. When F is constant, $F(G_{n,w}) = F$ and the lemma trivially holds. When F is a non-constant sparse-substituted input with t terms, $F(G_{n,w})$ is non-constant for $w > \lceil \log t \rceil + 1$ by Lemma 2.15. In the induction step F is non-constant and not a sparse-substituted input. There are two induction cases.

Case (i): The top gate of F is an addition gate: $F = \sum_{i=1}^{m} F_i$, where the F_i's are structurally-multilinear sparse-substituted depth-$(D-1)$ read-once formulae. If F has only one non-constant branch, say F_1, the induction hypothesis implies that $F_1(G_{n,w})$ is non-constant and hence $F(G_{n,w})$ is non-constant. Otherwise, assume that the branches F_1 and F_2 are non-constant. Then, because F is read-once: F_1 and F_2 are variable disjoint, without loss of generality $|\mathrm{var}(F_1)| \leq \frac{|\mathrm{var}(F)|}{2}$, and for any $x \in \mathrm{var}(F_1)$ there exists

$\gamma \in \bar{\mathbb{F}}$ such that $\partial_{x,\gamma} F = \partial_{x,\gamma}(\sum_{i=1}^{m} F_i) = \partial_{x,\gamma} F_1 \not\equiv 0$. Thus, $\partial_{x,\gamma} F$ has depth at most $D-1$ and depends on at most $\frac{|\mathrm{var}(F)|}{2}$ variables. Observe that

$$\min\left\{\left\lceil \log \frac{|\mathrm{var}(F)|}{2} \right\rceil, D-1\right\} + \lceil \log t \rceil + 1 = w - 1.$$

The induction hypothesis immediately gives that the $\partial_{x,\gamma} F \not\equiv 0$ is hit by $G_{n,w-1}$. Applying Lemma 2.14 implies that $F(G_{n,w-1}+G_{n,1})$ is non-constant. By the Proposition 2.12, Part (iii), $G_{n,w-1}+G_{n,1} = G_{n,w}$, completing this case.

Case (ii): The top gate of F is a multiplication gate: $F = \prod_{i=1}^{m} F_i$, where the F_i are structurally-multilinear sparse-substituted depth-$(D-1)$ read-once formulae. The induction hypothesis immediately implies that $G_{n,w'}$ hits each F_i, where $w' \doteq \min\{\lceil \log |\mathrm{var}(F)| \rceil, D-1\} + \lceil \log t \rceil + 1$. Further, at least one F_i must be non-constant and each $F_i(G_{n,w'})$ is non-constant if F_i non-constant. Combining this with the fact that $w \geq w'$ implies that $F(G_{n,w})$ is non-constant, completing this case. □

We formally conclude using Lemmas 4.4, 5.9, and 6.2 to prove the following main result.

Theorem 6.3. *For some function $w_k = k^{O(k)}$, the polynomial map $G_{n,w_k\cdot(\log(t)+1)+k\log n}$ is a hitting set generator for structurally-multilinear sparse-substituted formulae, where n denotes the number of variables, k the maximum number of substitutions in which a variable appears, and t the maximum number of terms a substitution consists of.*

Proof. We proceed by induction on k and argue that we can set w_k equal to the value $w_{2,k}$ from Lemma 5.9. The base case is immediate from Lemma 6.2. Consider the induction step for arbitrary k. Assume that $\mathcal{G} \doteq G_{n,w_k\cdot(\log(t)+1)+k\log n}$ is a generator for structurally-multilinear sparse-substituted read-k formulae. Lemma 5.9 with $m = 2$ implies that $\mathcal{G} + G_{n,w_k\cdot(\log(t)+1)}$ is a generator for structurally-multilinear sparse-substituted Σ^2-read-k formulae. Apply Lemma 4.4 to $\mathcal{G}' \doteq \mathcal{G} + G_{n,w_k\cdot(\log(t)+1)}$. This gives that $G_{n,w_k\cdot(\log(t)+1)+k\log n} + G_{n,w_k\cdot(\log(t)+1)} + G_{n,\log n}$ is a generator for structurally-multilinear read-$(k+1)$ formulae. Apply the basic properties of the SV-generator from Proposition 2.12, Part (iii), to get that a total seed length of $2w_k \cdot (\log(t)+1)+(k+1)\log n$ suffices to hit structurally-multilinear read-$(k+1)$ formulae. Observe that $2w_k \leq w_{k+1}$, and the theorem follows. □

A structurally-multilinear formula F on n variables, with individual degree d, has total degree at most dn. The SV-generator $G_{n,w}$ with output length n has total degree at most n. Combining these facts and Proposition 2.10 with Theorem 6.3 establishes the following.

Theorem 6.4 (Main Result – Blackbox). *There exists a deterministic blackbox polynomial identity test for structurally-multilinear sparse-substituted formulae that runs in time*

$$(dn)^{k^{O(k)}(\log(t)+1)+O(k\log n)}$$

and queries points from an extension field of size $O(dn^2)$, where n denotes the number of variables, k the maximum number of substitutions in which a variable appears, t the maximum number of terms a substitution consists of, and d the maximum degree of individual variables in the substitutions.

This theorem instantiates to the blackbox part of Theorem 1.2 when k is constant and further to the blackbox part of Theorem 1.1 when $t = d = 1$.

6.3. Special Case of Constant-Depth. We can improve the running time of our blackbox constant-read identity test by further restricting formulae to be constant-depth. We consider only the blackbox case because that is where we can get a substantial improvement. In the constant-depth setting we allow addition and multiplication gates that have arbitrary fanin. In order to specialize our previous argument to the constant depth case, we first give a version of the structurally-multilinear Fragmentation Lemma (Lemma 3.6) parameterized with respect to the depth. We then carry through the different parameterization in Lemma 4.4 and Theorem 6.3.

LEMMA 6.5 (Bounded-Depth Fragmentation Lemma). *Let V be a non-empty set of variables, $k \geq 2$, $D \geq 1$, and F be a depth-D structurally-multilinear sparse-substituted read$_V$-k formula such that $V \subseteq \operatorname{var}(F)$. Let t denote the maximum number of terms in each substituted polynomial. There exists a variable $x \in V$ and $\alpha \in \bar{\mathbb{F}}$ such that $\partial_{x,\alpha} F$ is non-zero and is*

(i) either the product of formulae of depth at most $D-1$, or else

(ii) a single Σ^k-read$_V$-$(k-1)$ formula.

Moreover, in each case the factors are of the form g or $\partial_{x,\alpha} g$ where g is a subformula of F.

Note the stronger either/or format of the Bounded-Depth Fragmentation Lemma in comparison with the general Fragmentation Lemma (Lemma 3.6). Another difference is that the proof of the former is straightforward.

PROOF. First note that since $V \subseteq \operatorname{var}(F)$, for each $x \in V$ there is an $\alpha \in \bar{\mathbb{F}}$ with $\partial_{x,\alpha} F \not\equiv 0$. Pick any such α (for the x we will select below).

There are two cases based on the output gate of F. Consider the case where $F = \prod_i g_i$. In this case F is already the product of subformulae that have depth at most $D-1$, and by structural multilinearity the same holds for every directional partial derivative of F. This completes the first case.

Consider the case where $F = \sum_i g_i$. Suppose g_1 contains k occurrences of some $x \in V$. Then $\partial_{x,\alpha} F \equiv \partial_{x,\alpha} g_1$ is a depth $D-1$ formula. Otherwise, no g_i contains k occurrences of any individual variable in V, so F itself is a Σ^k-read$_V$-$(k-1)$ formula, and so is every directional partial derivative of F. Selecting any $x \in V$ does the job then. This completes the second case. □

Lemma 6.5 leads to the following variant of Lemma 4.4 in the bounded-depth setting.

LEMMA 6.6. *For an integer $k \geq 1$, let $\mathcal{G}$ be a generator for n-variate structurally-multilinear sparse-substituted depth-D Σ^{k+1}-read-k formulae and let F be a non-zero n-variable structurally-*

multilinear sparse-substituted depth-D read-$(k+1)$ formula. Then $\mathcal{G} + G_{n,D}$ hits F.

Proof. First observe that if F is read-k, we are immediately done because $F(\mathcal{G}) \not\equiv 0$ and $\bar{0}$ is in the image of the SV-generator (by the first item of Proposition 2.12).

The proof goes by induction on D. If $D = 0$, the lemma holds trivially as F is constant. If $D = 1$, F is a read-once formula, which is covered by the above observation. For the induction step, by the above observation we can assume that F is read-$(k+1)$ and not read-k. Therefore, F meets the conditions to apply the second part of the Fragmentation Lemma for bounded depth formulae (Lemma 6.5). The lemma produces a variable $x \in \mathrm{var}(F)$ and $\alpha \in \bar{\mathbb{F}}$. The factors of $\partial_{x,\alpha} F$ all have depth at most $D-1$ and are structurally-multilinear read-$(k+1)$ formulae, except for at most one which might be a Σ^{k+1}-read-k formula. The induction hypothesis gives that the former factors of $\partial_{x,\alpha} F$ are all hit by $\mathcal{G} + G_{n,D-1}$. The latter factor (if it occurs) is hit by $\mathcal{G}$. Applying Lemma 2.14 gives that $\mathcal{G} + G_{n,D-1} + G_{n,1}$ hits F. Recalling Proposition 2.12, Part (iii), implies that $\mathcal{G} + G_{n,D}$ hits F. □

We can use the previous lemma with Lemmas 5.9 and 6.2 to construct a hitting set generator specialized to bounded depth. The proof is almost identical to Theorem 6.3, except that fanin of the reduced instance increases to $k+1$ from 2. This weakens the parameterization of the seed length with respect to k.

Theorem 6.7. *For some function $w_k = k^{O(k^2)}$, the polynomial map $G_{n,w_k \cdot (\log(t)+1)+kD}$ is a hitting set generator for structurally-multilinear sparse-substituted depth-D formulae, where n denotes the number of variables, D the depth of the formula, k the maximum number of substitutions in which a variable appears, and t the maximum number of terms a substitution consists of.*

Proof. We proceed by induction on k and argue that we can set w_k equal to the value $w_{k+1,k}$ from Lemma 5.9. The base case is immediate from Lemma 6.2. Consider the induction step for arbitrary k. Assume that $\mathcal{G} \doteq G_{n,w_k \cdot (\log(t)+1)+kD}$ is a generator for structurally-multilinear depth-D read-k formulae.

Lemma 5.9 with $m = k+1$ implies that $\mathcal{G} + G_{n,w_k\cdot(\log(t)+1)}$ is a generator for structurally-multilinear depth-D Σ^{k+1}-read-k formulae. Apply Lemma 6.6 to $\mathcal{G}' \doteq \mathcal{G} + G_{n,w_k\cdot(\log(t)+1)}$. This gives that $G_{n,w_k\cdot(\log(t)+1)+kD} + G_{n,w_k\cdot(\log(t)+1)} + G_{n,D}$ is a generator for structurally-multilinear depth-D read-$(k+1)$ formulae. Apply the basic properties of the SV-generator from Proposition 2.12, Part (iii), to get that a total seed length of $2w_{k+1}\cdot(\log(t)+1)+(k+1)D$ suffices to hit structurally-multilinear depth-D read-$(k+1)$ formulae. Observe that $2w_k \leq w_{k+1}$, and the theorem follows. □

Analogous to the unbounded-depth setting, combining Theorem 6.7 with Proposition 2.10 establishes the following theorem.

Theorem 6.8 (Improvement for Bounded-Depth Formulae). *There exists a deterministic blackbox polynomial identity test for structurally-multilinear sparse-substituted formulae with unbounded fanin that runs in time*

$$(dn)^{k^{O(k^2)}(\log(t)+1)+O(kD)}$$

and queries points from an extension field of size $O(dn^2)$, where n denotes the number of variables, D the depth of the formula, k the maximum number of substitutions in which a variable appears, t the maximum number of terms a substitution consists of, and d the maximum degree of individual variables in the substitutions.

The important difference between the above theorem and Theorem 6.4 is that the exponent no longer depends on n. When the read of a formula is constant we obtain the following corollary.

Corollary 6.9. *There exists a deterministic blackbox polynomial identity test for structurally-multilinear sparse-substituted constant-depth constant-read formulae that runs in time $(dn)^{O(\log t)}$ and queries points from an extension field of size $O(dn^2)$, where n denotes the number of variables, t the maximum number of terms a substitution consists of, and d the maximum degree of individual variables in the substitutions.*

Additionally, if the sparsity of substituted polynomials is constant the algorithm runs in polynomial time. In particular, we obtain the following corollary.

Corollary 6.10. *There is a deterministic polynomial-time blackbox identity test for multilinear constant-depth constant-read formulae.*

Note that in much of the prior work the term "constant-depth circuit" is used instead of "constant-depth formula." The two notions are equivalent in the sense that the standard transformation between circuits and formulae preserves the depth while yielding only a polynomial blow-up in the size. See (Raz & Yehudayoff 2008; Shpilka & Yehudayoff 2010) for further discussion. In general this transformation does not preserve the read value. In fact, it is meaningless to define "read-k circuits" since in a circuit the fanout of a gate is unbounded. However, in the cases we consider the read value is never affected.

Recently, Agrawal *et al.* (2012) used the notion of algebraic dependence and presented a unified approach for obtaining identity tests for constant-depth constant-read formulae. In fact, they develop a polynomial-time blackbox identity test for constant-depth constant-read formulae with addition, multiplication, and powering gates,[2] without the restriction of multilinearity. However, their technique requires unbounded field characteristic and thus only subsumes Theorem 6.8 (and other constant-depth constant-read results (Karnin *et al.* 2013; Saraf & Volkovich 2011; Shpilka & Volkovich 2009)) in the case of characteristic zero fields.

Acknowledgements

The authors would like to thank Amir Shpilka for bringing them in touch with each other, and the anonymous reviewers for their comments.

The first two authors were partially supported by NSF grants CCF-0728809, CCF-1017597, and CCF-1319822. The third author was partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575.

[2] In Agrawal *et al.* (2012) such formulae are referred to as "constant-depth constant-*occur*" formulae.

A preliminary version of this paper appeared in Anderson *et al.* (2011).

References

S. Aaronson & D. van Melkebeek (2011). On Circuit Lower Bounds from Derandomization. *Theory of Computing* **7**(1), 177–184.

M. Agrawal (2003). On derandomizing tests for certain polynomial identities. In *Proceedings of the 18th Annual IEEE Conference on Computational Complexity*, 355–359. ISBN 0769518796. ISSN 1093-0159.

M. Agrawal & S. Biswas (2003). Primality and identity testing via chinese remaindering. *Journal of the ACM* **50**(4), 429–443.

M. Agrawal, C. Saha, R. Saptharishi & N. Saxena (2012). Jacobian hits circuits: Hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, 599–614.

M. Agrawal & V. Vinay (2008). Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual Symposium on Foundations of Computer Science*, 67–75.

M. Anderson, D. van Melkebeek & I. Volkovich (2011). Derandomizing Polynomial Identity Testing for Multilinear Constant-Read Formulae. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, 273–282.

V. Arvind & P. Mukhopadhyay (2010). The ideal membership problem and polynomial identity testing. *Information and Computation* **208**(4), 351–363.

W. Baur & V. Strassen (1983). The complexity of partial derivatives. *Theoretical Computer Science* **22**, 317–330.

M. Ben-Or & P. Tiwari (1988). A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, 301–309. ISBN 0897912640.

M. Bläser, M. Hardt, R. Lipton & N. Vishnoi (2009). Deterministically testing sparse polynomial identities of unbounded degree. *Information Processing Letters* **109**(3), 187–192.

R. DeMillo & R. Lipton (1978). A probabilistic remark on algebraic program testing. *Information Processing Letters* **7**(4), 193–195.

Z. Dvir & A. Shpilka (2007). Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing* **36**(5), 1404–1434. ISSN 0097-5397.

Z. Dvir, A. Shpilka & A. Yehudayoff (2009). Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM Journal on Computing* **39**(4), 1279–1293.

A. Gupta, P. Kamath, N. Kayal & R. Saptharishi (2013). Arithmetic circuits: A chasm at depth three. Technical Report 26, Electronic Colloquium on Computational Complexity.

M. Jansen, Y. Qiao & J. Sarma (2009). Deterministic Identity Testing of Read-Once Algebraic Branching Programs. Technical Report abs/0912.2565, CoRR.

V. Kabanets & R. Impagliazzo (2004). Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity* **13**(1), 1–46.

Z. Karnin, P. Mukhopadhyay, A. Shpilka & I. Volkovich (2013). Deterministic Identity Testing of Depth 4 Multilinear Circuits with Bounded Top Fan-In. *SIAM Journal on Computing* **42**(6), 2114–2131.

Z. Karnin & A. Shpilka (2008). Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity*, 280–291.

N. Kayal & S. Saraf (2009). Blackbox polynomial identity testing for depth 3 circuits. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 198–207.

J. Kinne, D. van Melkebeek & R. Shaltiel (2012). Pseudorandom Generators, Typically-Correct Derandomization, and Circuit Lower Bounds. *Computational Complexity* **21**(1), 3–61.

A. Klivans & A. Shpilka (2006). Learning restricted models of arithmetic circuits. *Theory of computing* **2**(10), 185–206.

A. Klivans & D. Spielman (2001). Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, 216–223.

L. Lovász (1979). On determinants, matchings and random algorithms. In *Fundamentals of Computation Theory*, volume 79, 565–574.

T. Mignon & N. Ressayre (2004). A Quadratic Bound for the Determinant and Permanent Problem. *International Mathematics Research Notices* **79**, 4241–4253.

N. Nisan & A. Wigderson (1996). Lower bound on arithmetic circuits via partial derivatives. *Computational Complexity* **6**, 217–234.

R. Raz & A. Yehudayoff (2008). Lower Bounds and Separations for Constant Depth Multilinear Circuits. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity*, 128–139.

S. Saraf & I. Volkovich (2011). Black-Box Identity Testing of Depth-4 Multilinear Circuits. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing*, 421–430.

N. Saxena (2008). Diagonal circuit identity testing and lower bounds. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, 60–71.

N. Saxena (2009). Progress on polynomial identity testing. *Bulletin of the EATCS* **99**, 49–79.

N. Saxena & C. Seshadhri (2009). An almost optimal rank bound for depth-3 identities. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, 137–148.

N. Saxena & C. Seshadhri (2010). From Sylvester-Gallai Configurations to Rank Bounds: Improved Black-Box Identity Test for Depth-3 Circuits. In *Proceedings of the 51st Annual Symposium on Foundations of Computer Science*, 21–29.

N. Saxena & C. Seshadhri (2012). Blackbox Identity Testing for Bounded Top-Fanin Depth-3 Circuits: The Field Doesn't Matter. *SIAM Journal on Computing* **41**(5), 1285–1298.

J. Schwartz (1980). Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM* **27**(4), 701–717.

A. Shpilka & I. Volkovich (2008). Read-once polynomial identity testing. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, 507–516.

A. Shpilka & I. Volkovich (2009). Improved polynomial identity testing for read-once formulas. In *Proceedings of the 13th International Workshop on Randomization and Computation*, 700–713.

A. Shpilka & A. Wigderson (2001). Depth-3 Arithmetic Circuits over Fields of Characteristic Zero. *Computational Complexity* **10**(1), 1–27.

A. Shpilka & A. Yehudayoff (2010). Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science* **5**(3–4), 207–388.

R. Zippel (1979). Probabilistic algorithms for sparse polynomials. *Symbolic and Algebraic Computation* 216–226.

Manuscript received 28 June 2013

Matthew Anderson
University of Wisconsin-Madison
mwa@cs.wisc.edu

Dieter van Melkebeek
University of Wisconsin-Madison
dieter@cs.wisc.edu

Ilya Volkovich
Technion
ilyav@cs.technion.ac.il