Random Arithmetic Formulas can be Reconstructed Efficiently

Full Version

Ankit Gupta * Neeraj Kayal † Youming Qiao ‡

November 30, 2012

Abstract

Informally stated, we present here a randomized algorithm that given blackbox access to the polynomial f computed by an unknown/hidden arithmetic formula ϕ reconstructs, *on the average*, an equivalent or smaller formula $\hat{\phi}$ in time polynomial in the size of its output $\hat{\phi}$.

Specifically, we consider arithmetic formulas wherein the underlying tree is a complete binary tree, the leaf nodes are labelled by affine forms (i.e. degree one polynomials) over the input variables and where the internal nodes consist of alternating layers of addition and multiplication gates. We call these alternating normal form (ANF) formulas. If a polynomial f can be computed by an arithmetic formula μ of size s, it can also be computed by an ANF formula ϕ, possibly of slightly larger size $s^{O(1)}$. Our algorithm gets as input blackbox access to the output polynomial f (i.e. for any point $\mathbf{x}$ in the domain, it can query the blackbox and obtain $f(\mathbf{x})$ in one step) of *a random ANF formula ϕ of size s* (wherein the coefficients of the affine forms in the leaf nodes of ϕ are chosen independently and uniformly at random from a large enough subset of the underlying field). With high probability (over the choice of coefficients in the leaf nodes), the algorithm efficiently (i.e. in time $s^{O(1)}$) computes an ANF formula $\hat{\phi}$ of size s computing f. This then is the strongest model of arithmetic computation for which a reconstruction algorithm is presently known, albeit efficient in a distributional sense rather than in the worst case.

*Microsoft Research India, t-ankitg@microsoft.com

†Microsoft Research India, neeraka@microsoft.com

‡Institute for Interdisciplinary Information Sciences, Tsinghua University, jimmyqiao86@gmail.com

Contents

1 Introduction

What is the smallest formula computing a given multivariate polynomial f? One way to understand and shed light on this problem is via the investigation of the computational complexity of the corresponding algorithmic problem known as the reconstruction problem. A reconstruction algorithm for a polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$ is given blackbox access to f and is required to output a (succinct) representation of f in some suitable model of computation such as arithmetic formulas. The algorithm can adaptively query the blackbox to evaluate f on inputs of its choice from $\mathbb{F}^n$. [1] The algorithm is said to be efficient, if the running time, and hence number of queries, is bounded by a polynomial in the size of the representation produced by it. The most obvious representation of a multivariate polynomial is its formula as a sum, weighted by coefficients from $\mathbb{F}$, of monomials, i.e., a depth-2 $\Sigma\Pi$ formula. In this case, the problem of reconstruction is more commonly referred to as *interpolation*. In the past few decades, the interpolation problem has drawn considerable attention (see e.g. [BOT88, KY88, Zip90, KS01] and the references therein) and found many applications as well. Of course, writing down all the monomials of a polynomial may be too expensive. For example, the polynomial $\sum_{S \subseteq [n]} (\prod_{i \in S} X_i) \times (\prod_{i \in ([n] \setminus S)} Y_i)$ can also be written as $(X_1 + Y_1) \times (X_2 + Y_2) \times \ldots \times (X_n + Y_n)$. It is seen that the first one (sum of monomials) involves $(n-1) \cdot (2^n - 1)$ operations. On the other hand, the second expression only needs $(2n-1)$. Of course, the second expression is also natural, and has been employed implicitly by mathematicians. The technical term for such an expression is an *arithmetic formula*[2], and the size of a formula is the number of operations involved. The naive example above indicates that it is more desirable to solve the reconstruction problem with the output polynomials represented by a small formula. We say the size of the smallest formula computing f is the *formula size* of f, and use $\mathsf{fs}(f)$ to denote this quantity. Ideally, we would like a reconstruction algorithm, given black-box to an n-variate polynomial f, outputs a formula computing it in time polynomial in n, and $\mathsf{fs}(f)$ [3]. We call this version of the reconstruction problem, the *arithmetic formula reconstruction problem*.

More generally, we can consider reconstruction problems for classes of arithmetic circuits. From a broad perspective, reconstructing polynomials from arithmetic complexity classes is, in some sense, analogous to learning concept classes of Boolean functions using membership and equivalence queries. (see Chapter 5 of the survey by Shpilka and Yehudayoff [SY10] for justifying arguments for the analogy to the Boolean world and, more generally, for previous work in this area.) While research on the theory of learnability in the Boolean world has evolved into a mature discipline, thanks to fundamental notions such as PAC learning due to Valiant, research on learnability in the arithmetic world has been gaining momentum only in recent years.

Hardness of reconstruction. The reconstruction problem, in its most general formulation, e.g. produce (roughly) an optimal arithmetic circuit (resp. Boolean circuit) for a given polynomial (resp. Boolean function) f, seems to be extremely hard. An informal but intuitive reason is that to solve this problem, conceptually the algorithm should somehow "know" the size of the smallest formula for every polynomial (resp. Boolean function)! On the other hand the status of the field of arithmetic complexity is such that we do not know of a superquadratic lower bound for any explicit polynomial. Formal hardness results are available for restricted subclasses of Boolean and arithmetic circuits [4].

[1] We typically assume f itself to have a small representation in the target model of computation. This is essentially without loss of generality because of the DeMillo-Lipton-Schwarz-Zippel lemma which allows us to efficiently test whether the output of the reconstruction algorithm represents f or not.

[2] An *arithmetic formula* is a rooted tree with two types of internal nodes/gates: a $\times$ gate computes the product of its inputs, $+$ gate computes an arbitrary linear combination of its inputs, wires are labelled by elements of a field $\mathbb{F}$ and leaves by elements of $\mathbf{X} \cup \mathbb{F}$, where $\mathbf{X}$ is a set of variables.

[3] Note that the running time of a reconstruction algorithm is an upper bound on the size of the representation output by it. Note also that the formula size of f is an upper bound on its degree.

[4] Indeed it seems that even proving the hardness of reconstruction itself requires proving some extremely weak but still

Specifically, Allender et al [AHM+08] showed that finding even an approximately optimal DNF formula is NP-hard, even when the Boolean function is given rather verbosely as a truth table. Buchburger and Umans [BU08] showed that when the boolean function is given succinctly as a formula then finding an equivalent minimal formula is Σ_2^{P}-complete (i.e. hard even for the second level of the polynomial hierarchy).[5] In the arithmetic setting, Håstad [Hås90] showed that reconstructing the smallest depth three set-multilinear formula (a much weaker model than general arithmetic formulas) for a given set-multilinear polynomial is NP-hard even for degree three polynomials.

Arithmetic formula reconstruction problem in average case. These hardness results indicate that it may be unrealistic to hope for an efficient worst-case reconstruction algorithm even when the polynomial (resp. Boolean function) f is given rather verbosely as a $\Sigma\Pi$ formula (resp. as a truth table). Moreover since the formula-size of a polynomial is still poorly understood, it seems difficult to make progress on the formula reconstruction problem. However, it turns out that we are able to solve an average-case version of this problem. Before stating our result more precisely we mention some related work. Because of its hardness, progress on the reconstruction problem has been possible only for very restricted classes of arithmetic (resp. Boolean) formulas and/or under distributional assumptions (i.e. average-case rather than worst-case). We now mention some of the known results of this flavor.

Previous works. In the Boolean world an algorithm for reconstructing *random DNF formulas* was given recently by Sellie [Sel09]. Another line of work pertaining to learning DNF formulas stems from a conjecture by Mansour [Man94] which was recently settled for random DNF formulas by Klivans, Lee and Wan [KLW10]. Reconstruction/learning in the arithmetic setting has gained momentum only recently and we mention some of this work here. Reconstruction algorithms are previously known for depth-2 $\Sigma\Pi$ formulas (sparse polynomials) [KS01], *read-once* arithmetic formulas [SV08, BC98], non-commutative *arithmetic branching programs* [AMS10], $\Sigma\Pi\Sigma(2)$ formulas, i.e., depth-3 formulas [6] with top + gate of fan-in 2, [Shp07] [7], and $\Sigma\Pi\Sigma(k)$ formulas with $k = O(1)$ [KS09] [8] For more information on reconstruction in the arithmetic setting and previous work in the area, we refer the reader to chapter 5 of the survey by Shpilka & Yehudayoff [SY10]. More recently, Kayal [Kay12] and Gupta, Kayal and Lokam [GKL12] have devised reconstruction algorithms for some more classes of bounded-depth formulas. In [GKL11], Gupta, Kayal and Lokam have devised a reconstruction algorithm for multilinear formulas under distributional assumptions.

The distribution on input instances. An average-case version of a problem will assume some natural distribution over the input instances, and an algorithm for this is expected to work correctly for most instances sampled from this distribution. To come up with a natural distribution on arithmetic formulas seems tricky [9]. There is a very natural canonical form for arithmetic formulas which we call Alternating Normal Form (ANF for short) that we define below. Our reconstruction algorithm is then efficient on the average with respect to the natural uniform distribution on formulas in this normal form.

Formulas in Normal Form. We say that an arithmetic formula ϕ is an ANF formula if

1. The underlying tree of ϕ is a complete rooted binary tree (the root node is called the output

nontrivial circuit lower bounds [BU08] and this is perhaps the reason that we do not have hardness results for general arithmetic or Boolean circuits.

[5] Some further evidence of the hardness of reconstruction in the Boolean world is provided by Kabanets and Cai [KC00] and Fortnow and Klivans [FK06].

[6] Also some of these algorithms only work under some very mild and easily verifiable non-degeneracy conditions.

[7] The algorithms of [Shp07] and [KS09] have quasipolynomial time complexity.

[8] Also [BBB+00, KS03] devise an algorithm that given the polynomial computed by a hidden *set-multilinear* depth-3 formula, outputs an algebraic branching program/multiplicity automata of roughly the same size.

[9] The reason mostly lies on the fact that the internal tree structure of a formula (see Section 3.2) can be quite arbitrary. Here we fix the internal tree to be a complete binary tree.

node) [10], and

2. The internal nodes consist of alternating layers of + and × gates [11], and

3. The leaves of the tree are labelled with affine forms. i.e. each leaf is labelled with $\ell = a_0 + a_1X_1 + \ldots + a_nX_n$, where each $a_i \in \mathbb{F}$ is a scalar.

The proposition below motivates the consideration of ANF-formulas by noting that the size of the smallest ANF-formula computing any polynomial f is comparable to that of the smallest formula for it. Thus, as far as understanding the formula-size complexity of a polynomial is concerned, we can focus our attention on formulas in ANF form with only a relatively small loss in the quality of the answer.

Proposition (Proposition 5 in Section 3.2)**.** *Let $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be a polynomial computed by an arithmetic formula ϕ of size s. Then there exists an ANF formula $\hat{\phi}$ of size at most s^4 computing the same polynomial $f(\mathbf{X})$.*

See Section 3.2 for a proof. Having fixed the tree structure of the formula, there is a natural uniform distribution on ANF formulas wherein the coefficients in the affine forms at the leaves are picked uniformly at random from (a subset $S \subseteq \mathbb{F}$ of) the underlying field $\mathbb{F}$. We make this precise as follows. We say that an ANF formula ϕ is a $(\mathbf{X}, \Delta, S)$-ANF formula if it has depth 2Δ and for each linear form labelling

$$\ell = a_0 + a_1X_1 + a_2X_2 + \ldots + a_nX_n$$

a leaf node of ϕ, each coefficient a_i is in $S \subseteq \mathbb{F}$. In what follows S will typically be a finite set and we will consider the uniform distribution on the set of $(\mathbf{X}, \Delta, S)$-ANF formulas [12]. For ANF-formulas of size s, let us denote the above P-samplable distribution by $\mathcal{D}(n, s, S)$. Now we give the *average case version* of the formula reconstruction problem with respect to $\mathcal{D}(n, s, S)$: assume we are given black-box access to an ANF-formula ϕ sampled from $\mathcal{D}(n, s, S)$, and ϕ computes $f \in \mathbb{F}[X_1, \ldots, X_n]$. Our goal is to construct an arithmetic formula $\hat{\phi}$ computing f, in time $\mathsf{poly}(n, s)$, with high probability over $\mathcal{D}(n, s, S)$ [13].

Our contribution. We present here an efficient randomized algorithm that given access to the output of *a random ANF formula* can reconstruct the hidden underlying formula.

Theorem 1. *Let $\mathbb{F}$ be a field of characteristic 0 and S be a finite subset of $\mathbb{F}$. Assume there is a black box holding an ANF-formula ϕ of size s sampled from $\mathcal{D}(n, s, S)$, and ϕ computes a polynomial $f \in \mathbb{F}[X_1, \ldots, X_n]$. There is a randomized algorithm that given this black box, either outputs an ANF-formula $\hat{\phi}$ of size $\leq s$ computing f, or outputs* `Fail` *. The algorithm succeeds for $(1 - \frac{n^2.s^{O(1)}}{|S|})$ fraction of the ANF-formulas from $\mathcal{D}(n, s, S)$. Moreover, the running time of the algorithm is at most $(n \cdot s)^{O(1)}$.* [14]

This result then represents the strongest model of arithmetic computation for which a reconstruction algorithm is presently known, *albeit efficient in an average-case or distributional sense rather than*

[10] In particular $\text{size}(\phi) = 2^{\text{depth}(\phi)+1} - 1$, where $\text{size}(\phi)$ is the number of nodes in the tree of ϕ and $\text{depth}(\phi)$ is the maximum distance of a leaf node from the output node of ϕ.

[11] In particular the label of an internal node at distance d from the closest leaf node is + if d is even and × otherwise. So if the root node is a + node, its children are all × nodes, its grandchildren are all + etc.

[12] Abusing terminology, we will sometimes say that a random (ANF) formula has some property if at least $(1 - o_{|S|}(1))$ fraction of $(\mathbf{X}, \Delta, S)$-ANF formulas have that property.

[13] This also implies that $|\hat{\phi}|$ is $\mathsf{poly}(d, n, \mathsf{fs}(f))$.

[14] We ignore the internal randomness of the algorithm as by well-known techniques, the probability of failure with respect to the internal randomness can effectively be made negligible

in the worst case. The conceptual contribution of this paper is to show how the dimension of the singular locus of a polynomial - a property used in proving a moderate lower bound for the determinant versus permanent problem by von zur Gathen in [vzG87] - can be used to efficiently do reconstruction of random ANF formulas. The technical work involves analyzing the components of the singularities of arbitrary linear combinations of random formulas and to characterize the high dimensional ones among these.

A few remarks are due to the theorem itself.

Remark 2.

1. **Arithmetic Formulas versus Boolean formulas.** Note that a boolean formula over the basis $\{\oplus, \wedge, \neg\}$ is equivalent to arithmetic formulas over the finite field $\mathbb{F}_2$. However our algorithm *cannot be used for boolean formula reconstruction* as the algorithm inherently requires the underlying field to be larger than the formal degree of the formula. On the other hand, as long as the size of the field itself is large enough, theorem 1 continues to remain valid even over fields with characteristic two. A few changes to the analysis are however required in the low characteristic case and we point these out in more detail in remark 68 after the proof of this theorem.

2. **Time complexity and number of arithmetic operations.** For ease of presentation we will assume throughout the rest of this paper that the elementary field operations $(+, -, \times, \div)$ and the extraction of roots of constant degree polynomials are all unit cost operations. Note that over a finite field $\mathbb{F}_q$ the elementary field operations as well as root extraction of constant degree polynomials can be done in actual time $(\log q)^{O(1)}$ so that overall the time complexity is still polynomial in terms of the bitlength of the hidden formula. Over the field of rational numbers one needs to be more careful. It turns out that the steps of our algorithm ultimately boil down to doing some linear-algebraic computation and extraction of roots of constant degree polynomials. Moreover the bitlength of the entries of the relevant matrices as well as the bitlength of the coefficients of the relevant polynomials is bounded by a polynomial in the size of the hidden formula so that our algorithm has polynomial running time in this case too. Abusing terminology, we will often that an algorithm has running time $t(n)$ when the number of arithmetic operations in that algorithm is $t(n)$.

3. **Smoothed Complexity.** By concatenating the vector of $(n+1)$ coefficients of the affine forms occuring in the 4^{Δ} leaf nodes of an ANF formula into one long vector of length $N = 4^{\Delta} \cdot (n+1)$, we can view the set of $(\mathbf{X}, \Delta)$-ANF formulas as points in the space $\mathbb{F}^N$. As we will see the input instances on which our algorithm fails forms a proper Zariski-closed subset [15] of $\mathbb{F}^N$.

   This means that *for any* ANF formula ϕ, our algorithm will succeed with high probability given access to the output polynomial of a "slightly perturbed" formula $\hat{\phi}$. In this sense our algorithm has smoothed polynomial-time complexity.

4. **Parallelization.** Our algorithm can easily be parallelized as most of the steps of the algorithm ultimately boil down to linear algebraic computations. In particular, *assuming that the underlying field operations and extraction of roots of a constant degree polynomial are all unit cost operations* our algorithm can be implemented in randomized NC.

5. **Limitations of our algorithm.** One can consider formulas wherein addition gates have larger fanin and while our algorithm can be generalized somewhat for such formulas, the running time degrades rapidly if the fanin of the addition gates is large. In particular this rules out the

[15] Recall that a Zariski-closed subset of $\mathbb{F}^n$ is the set of common zeroes of some system of polynomial equations over $\mathbb{F}$.

application of our algorithm for the computation of the size of the smallest formula for (small instances of) matrix multiplication [16].

1.1 Discussion

Average-case algorithms, by their very nature, are highly sensitive to the distribution from which the input instances are drawn. Unlike graphs, for which the Erdös-Renyi model of generating graphs (the so-called $G(n,p)$ graph) is pretty much a universally accepted notion of what constitutes a random graph, there is no obvious way to define a random formula. In this work we showed that for a reasonable notion of a random formula, reconstruction can be done efficiently. At the same time, the impact of this work on pratically important problems is not immediately clear. Perhaps the most important practical application by far of any formula reconstruction algorithm might be to the complexity of matrix multiplication. For example, a reconstruction algorithm producing optimal sized formula for the polynomial corresponding to the product of say two 5×5 matrices may potentially improve the current best asymptotic running time of matrix multiplication using Strassen's recursive approach. Unfortunately our line of research here (reconstruction of random ANF formulas) does not seem to be immediately applicable to matrix multiplication. This is because the polynomial corresponding to matrix multiplication has degree 3 whereas any reasonable notion of a random formula will almost certainly generate polynomials of much larger degree. On the other hand, it seems reasonable to hypothesize that if a given polynomial f admits a formula of size $(\deg(f))^{O(1)}$ than our results and techniques should help reconstruct the formula (with high probability). We mention here in passing that in many application areas such as scientific computing, finance, vision, graphics, etc. computers routinely operate on numerical data and a large chunk of the computation in these applications involves the arithmetic operations of addition, multiplication, subtraction and division. It might be interesting to see if these algorithmic techniques can help optimize some of these fragments of arithmetic computation. We now make some comments with an eye towards such potential applications. We note here that in its current form, our analysis suggests that even though our algorithm has polynomial running time, it is completely impractical. But it might well be that simply a better analysis (see remark 68 for suggestions in this direction) can lead to a better choice of parameters and therefore also a significant speedup, and bringing our algorithm much closer to having a real-world existence. On the theoretical side, it is quite easy to construct formulas (corresponding to low-rank tensors) for which our algorithm fails rather miserably. But perhaps it is inevitable that there exist easily samplable hard instances against any algorithm aspiring to do formula reconstruction efficiently. As we mentioned earlier, the available hardness results indicate that it would be unrealistic to hope for an efficient worst-case reconstruction algorithm for arithmetic formulas. In summary, therefore, the algorithm we present has its limitations; nevertheless we feel that our work does represent a small step forward towards understanding a central question in arithmetic complexity: *what is an optimal arithmetic formula computing a given polynomial f?*

2 Basic Idea and Approach

Let $\mathbb{F}$ be an algebraically closed field and let S be a subset of $\mathbb{F}$. Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$. Suppose we have blackbox access to the output polynomial f of a random $(\mathbf{X}, \Delta, S)$-ANF formula ϕ. By querying f at points of our choice, we want to recover ϕ. How do we do so? In order to do this, our first task is determining the nature of the output node, i.e. whether its a $+$ gate or a $\times$ gate. This is done by

[16] The problem of multiplying two $n \times n$ matrices is essentially equivalent to computing the polynomial $\mathrm{Trace}(X \cdot Y \cdot Z)$, where X, Y and Z are $n \times n$ matrices whose entries are distinct formal variables. Since this polynomial has degree just three, the best formula for it is very shallow - a depth three circuit with the addition gate at the output of this circuits having a large fanin

testing the irreducibility of f. If the output node is a $\times$ gate, then f must be reducible else there is a smaller ANF formula computing f. On the other hand for a random ANF formula with a $+$ gate at the top, the output is an irreducible polynomial with high probability (corollary 56). In this way, having determined the nature of the output node, via the (ir)reduciblity of f, we examine the two cases separately.

Case I: Output node is a $\times$ gate. This is the easy case. We factor f using Kaltofen's algorithm. Now it can happen (in rare circumstances) that the number of factors of f is larger than the number of children of the output node. For a generic (i.e. randomly chosen) ANF formula ϕ these two quantities will however be equal (corollary 56) so that Kaltofen's algorithm provides blackbox access to the two children of the output node. We then recursively compute the formulas for the two children.

Case II: Output node is a $+$ gate. This is the main case and almost all the work we do involves handling this case. We need to go one level deeper. As the two children of the output node are $\times$ gates so that the output polynomial f is of the form

$$f = f_1 \cdot f_2 + f_3 \cdot f_4. \tag{1}$$

Our aim will be to obtain blackbox access to the four 'grandchildren' f_1, f_2, f_3 and f_4. If we could do this then we can recursively find ANF formulas for the f_i's and we would be done. Before we find the f_i's, let us ask an easier question: can we even distinguish an f of the form (1) from a truly random n-variate polynomial of degree $d = \deg(f)$? We first observe that if $\mathbf{x}$ is any point on the variety

$$\mathtt{V}(\mathbf{f}) := \{\mathbf{x} \in \mathbb{F}^n : f_1(\mathbf{x}) = f_2(\mathbf{x}) = f_3(\mathbf{x}) = f_4(\mathbf{x}) = 0\} \subseteq \mathbb{F}^n$$

then it is also a singularity of f, denoted $\mathbf{x} \in \mathrm{Sing}(f)$, where $\mathrm{Sing}(f) \subseteq \mathbb{F}^n$ is defined as the common zeroes of the system of equations

$$f(\mathbf{X}) = \frac{\partial f}{\partial X_1}(\mathbf{X}) = \ldots = \frac{\partial f}{\partial X_n}(\mathbf{X}) = 0. \tag{2}$$

This can be seen as follows: differentiating equation (1) with respect to the variable X_i we get

$$\frac{\partial f}{\partial X_i} = f_1 \cdot \frac{\partial f_2}{\partial X_i} + f_2 \cdot \frac{\partial f_1}{\partial X_i} + f_3 \cdot \frac{\partial f_4}{\partial X_i} + f_4 \cdot \frac{\partial f_3}{\partial X_i} \tag{3}$$

For this and equation (1) we have that any $\mathbf{x} \in \mathtt{V}(\mathbf{f})$ is a zero of $f(\mathbf{X})$ and also of $\frac{\partial f}{\partial X_i}(\mathbf{X})$ for each $i \in [n]$. When the number of variables n is larger than 4, $\mathtt{V}(\mathbf{f})$ (and therefore also $\mathrm{Sing}(f)$) is likely to contain lots of points while $\mathrm{Sing}(g)$ is empty for most n-variate polynomials of degree d. *The main idea of this work is to exploit the structure of* $\mathrm{Sing}(f)$ *- in particular the fact that* $\mathtt{V}(\mathbf{f})$ *is a subset of* $\mathrm{Sing}(f)$ *- to recover* $\mathtt{V}(\mathbf{f})$ *and then with some more work to recover the actual grandchildren - viz. the* f_i*'s.* At this juncture, let us point out some of the difficulties that we face and/or the questions that we need to address in implementing this idea.

(Q1) **Emptiness of** $\mathtt{V}(f)$ - For an arbitary $\mathbf{f} = (f_1, f_2, f_3, f_4)$, it can happen that $\mathtt{V}(\mathbf{f})$ is empty even when the number of variables n is much larger than 4 - e.g. when $f_2(\mathbf{X}) = 1 + f_1(\mathbf{X})$. If $\mathtt{V}$ is empty then it might not be possible to extract information about the f_i's by looking at $\mathrm{Sing}(f)$.

(Q2) **Analyzing** $\mathrm{Sing}(f)$ **in a computationally efficient manner** - Even some very basic questions pertaining to varieties, such as emptiness of a given variety, are NP-hard/coNP-hard. How do we analyze the structure of $\mathrm{Sing}(f)$ in a computationally efficient manner?

(Q3) **Exsistence of points in** $\mathrm{Sing}(f) \setminus \mathtt{V}(\mathbf{f})$ - $\mathrm{Sing}(f)$ can in general contain many points other than those on $\mathtt{V}(\mathbf{f})$. For a given point $\mathbf{x} \in \mathrm{Sing}(f)$, how can we know if it is coming from $\mathtt{V}(\mathbf{f})$ or not?

(Q4) **Recovering f from $\mathtt{V}(\mathbf{f})$** - In general a variety can be written as the common zeroes of a set of polynomials $\mathbf{g} \subseteq \mathbb{F}[\mathbf{X}]$ in several different ways. Given such a $\mathbf{g}$, how do we recover the actual f_i's from it?

(Q5) **Uniqueness of the f_i's** - A closely related difficulty pertains to the uniqueness of the represenation (1) for f. Aside from the trivial ones such as $f = f \cdot 1 + x^d \cdot 0$ there can be several nontrivial ways of writing f in the form of equation (1). Indeed, for any '*non-trivial*' representation of the form (1), there are several more nontrivial ones such as

$$f = (\frac{f_1 + f_3}{2}) \cdot (f_2 + f_4) + (\frac{f_1 - f_3}{2}) \cdot (f_2 - f_4).$$

The difficulty with this nonuniqueness is that if we obtain 'an incorrect' representation of f, say

$$f = g_1 \cdot g_2 + g_3 \cdot g_4,$$

then the g_i need not be computable by ANF formulas smaller than ϕ and so recursively invoking the algorithm on the g_i's will lead to failure. Such failed recursive calls would be prohibitively expensive.

(Q6) **Representation of the f_i's** - Even if we could somehow recover the f_i's, how do we store/represent them in an efficient manner? More generally, how do we efficiently represent polynomials computed at the intermediate nodes? Clearly, the f_i's will typically have exponentially many monomials so that a $\Sigma\Pi$ representation of the f_i's is prohibitively expensive. We could use a '*blackbox representation*' in the manner of Kaltofen wherein some '*advice*' is stored. When the value of f_1 at a point $\mathbf{x} \in \mathbb{F}^n$ is needed, we use the advice to query f at $\mathsf{poly}(\deg(f) \cdot n)$ points and based on these answers and the advice compute $f_1(\mathbf{x})$. Even this is too expensive because at a depth Δ from the root node of the tree corresponding to ϕ, we will end up making $(\deg(f) \cdot n)^{O(\Delta)}$ queries to f, which is superpolynomial in the size of the formula ϕ.

We now indicate how we go about addressing each of these difficulties.

Tackling Q1. We take care of the first difficulty *by homogenizing the polynomial f and working in the projective closure of $\mathbb{F}^n$ instead*. Recall that the homogenization of a polynomial f is the homogeneous polynomial

$$\hat{f}(X_0, X_1, \ldots, X_n) := X_0^d \cdot f(\frac{X_1}{X_0}, \frac{X_2}{X_0}, \ldots, \frac{X_n}{X_0}).$$

Note that $\hat{f}$ [17] now admits a homogeneous formula $\hat{\phi}$ of product-depth Δ; $\hat{\phi}$ is obtained from ϕ simply by replacing the label of each leaf node of ϕ from the affine form

$$\ell = a_0 + a_1 X_1 + a_2 X_2 + \ldots + a_r X_r$$

by the corresponding linear form

$$\hat{\ell} = a_0 X_0 + a_1 X_1 + a_2 X_2 + \ldots + a_r X_r.$$

Note that $\hat{\phi}$ is now a homogeneous formula of product-depth Δ. Equation (1) now becomes

$$\hat{f} = \hat{f}_1 \cdot \hat{f}_2 + \hat{f}_3 \cdot \hat{f}_4$$

To avoid notational clutter we replace f and ϕ by $\hat{f}$ and $\hat{\phi}$ respectively so that our problem reduces to the situation where we want to construct a homogeneous ANF formula for a given homogeneous

[17] Later in proposition 62 we will see that given blackbox access to f we can evaluate $\hat{f}$ even at points for which $X_0 = 0$.

polynomial f. In this manner, we can assume that the f_i's in equation (1) are homogeneous polynomials and are therefore polynomial functions over the corresponding projective space $\mathbb{P}^n$. Intersections of varieties are much more nicely behaved in projective spaces. Indeed from fact 19 it follows that $\mathtt{V}(\mathbf{f}) \subseteq \mathbb{P}^n$ will always be of dimension at least $(n-4)$ and in particular will always be nonempty when $n \geq 4$. To see the usefulness of this let us note a lower bound.

Proposition 3. *For any $d \geq 2$ and $n \geq 5$, the polynomial*

$$g = X_1^d + X_2^d + \ldots + X_n^d$$

cannot be computed by a homogeneous ANF formula.

Proof. Suppose if possible there exists a homogeneous ANF formula ϕ computing g. g is irreducible so that we can assume without loss of generality that the output node of ϕ is a + gate. From the discussion above we must then have $\mathrm{codim}(\mathrm{Sing}(g)) \leq 4$. But it is easily verified that in this case, $\mathrm{Sing}(g)$ is empty as a projective variety thereby giving a contradiction. □

Tackling Q2 and Q3. From the above discussion we have that $\mathtt{V}(\mathbf{f})$ is a subvariety of $\mathrm{Sing}(f)$ of codimension at most 4. The key observation here which is the source of our computational efficiency is that even though $\mathrm{Sing}(f)$ may in general be a very complicated variety and be difficult to analyze, we are really only interested in $\mathtt{V}(\mathbf{f})$ *which is a subvariety of* $\mathrm{Sing}(f)$ *of bounded codimension.* Let us now consider the set of points in $\mathrm{Sing}(f) \setminus \mathtt{V}(\mathbf{f})$. Let us consider an arbitrary $\mathbf{x} \in \mathrm{Sing}(f) \setminus \mathtt{V}(\mathbf{f})$. From equations (2) and (3) we get that

$$\begin{bmatrix} \frac{\partial f_1}{\partial X_1}(\mathbf{x}) & \frac{\partial f_2}{\partial X_1}(\mathbf{x}) & \ldots & \frac{\partial f_4}{\partial X_1}(\mathbf{x}) \\ \frac{\partial f_1}{\partial X_2}(\mathbf{x}) & \frac{\partial f_2}{\partial X_2}(\mathbf{x}) & \ldots & \frac{\partial f_4}{\partial X_2}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial X_n}(\mathbf{x}) & \frac{\partial f_2}{\partial X_n}(\mathbf{x}) & \ldots & \frac{\partial f_4}{\partial X_n}(\mathbf{x}) \end{bmatrix} \cdot \begin{bmatrix} f_2(\mathbf{x}) \\ f_1(\mathbf{x}) \\ f_4(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} = \mathbf{0}$$

The $(n \times 4)$ matrix of the lhs of the above equation is known as the *Jacobian* and denoted $J(\mathbf{f}, \mathbf{X})$. The fact that $\mathbf{x} \notin \mathtt{V}(\mathbf{f})$ means that the (4×1) vector on the lhs of the above equation is nonzero which in turn means that $J(\mathbf{f}, \mathbf{x})$ has rank 3. The set of points $\mathbf{x} \in \mathbb{F}^n$ for which $J(\mathbf{f}, \mathbf{x})$ has rank 3 forms a variety W (the common zeroes of all 4×4 minors of $J(\mathbf{f}, \mathbf{X})$.) We observe (corollary 54) that when f_1, f_2, f_3 and f_4 are independently chosen random formulas then the following algebraic condition is satisfied with high probability:

$$(\mathrm{FI}): \ \dim(\mathtt{V}(\mathbf{f})) = (n-4) \ \text{ and } \ \dim(\mathrm{Sing}(f) \cap W) < (n-4).$$

We use this to tackle Q2 and Q3 by showing (in lemma 42) that if the assumption FI (short for formulaic independence (definition 37)) is satisfied then given blackbox access to $\mathrm{Sing}(f)$ we can *efficiently* recover a basis for the corresponding ideal

$$\mathfrak{I}(\mathbf{f}) := \langle f_1, f_2, f_3, f_4 \rangle \subseteq \mathbb{F}[\mathbf{X}].$$

Tackling Q4 and Q5. A significant part of the technical work in this paper is aimed at tackling questions Q4 and Q5. By examining the homogeneous components of the generators of $\mathfrak{I}(\mathbf{f})$, we can efficiently obtain a tuple of polynomials

$$\mathbf{h} = (h_1, h_2, h_3, h_4) \in (\mathbb{F}[\mathbf{X}])^4$$

such that

$$\begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{pmatrix} = A \cdot \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}, \tag{4}$$

where $A \in \mathbb{F}^{4\times 4}$ is an invertible matrix (lemma 43). In other words, we get the f_i's upto $\mathbb{F}$-linear combinations. We show that if the grandchildren of the f_i's satisfy a certain other algebraic nondegeneracy condition (definition 48) then for *any* $\mathbb{F}$-linear combination

$$h = a_1 f_1 + a_2 f_2 + a_3 f_3 + a_4 f_4 \in \mathbb{F}[\mathbf{X}]$$

in which at least two of the a_i's are nonzero, the dimension of $\mathrm{Sing}(h)$ is at most $(n-5)$ (lemma 49). This other algebraic condition roughly captures independence of the polynomials which are the grandchildren of the f_i's and as before, this too is satisfied with high probability when ϕ is a randomly chosen formula (corollary 55). This then allows us to set up a system of polynomial equations in the unknown coefficients a_1, a_2, a_3, a_4 such that the solutions of this system correspond to linear combinations of the h_i's which are scalar multiples of the f_i's (lemma 24). It is well known that a system of polynomial equations [18] in a bounded number of unknowns can be solved efficiently (theorem 25). This allows us to recover (upto scalar multiples and permutation) the f_i's (lemma 51).

Tackling Q6. To overcome this difficulty we invoke what we call *the project and lift technique* (originally due to Kaltofen [Kal85] and Shpilka [Shp07]). While there is some nontrivial work needed to apply this to our setting let us give here an informal gist of the technique as applicable to our situation. We effectively reduce the problem to the situation where n, the number of variables is a constant, say r. This is done by judiciously choosing a small ($O(n)$) number of r-dimensional subspaces U of the underlying space $\mathbb{F}^n$ and doing the reconstruction for the restriction to each subspace U. With the number of variables reduced to $r = O(1)$ we can represent the polynomial computed at each intermediate node of the tree in the naïve representation. We then show how to patch together the ANF formulas for f restricted to these subspaces to obtain a formula for f over the original n-dimensional space (lemma 65).

Overview of the algorithm and its analysis. We now give the overall structure of our algorithm and its analysis keeping the above ideas in mind. The main component of our algorithm is a subroutine (algorithm 1) that we call the low dimensional reconstruction algorithm (`LDR` for short) wherein the input consists of a homogeneous $(r+1)$-variate polynomial f (r is a suitable constant) and the algorithm reconstructs the formula for f. For a technical reason [19] the output of `LDR` consists of the quadratic polynomials at the second-last layer of the tree rather than the affine forms at the leaves. The rest of the algorithm (algorithm 2) makes $O(n)$ invocations to `LDR` and patches the answers together to get a formula for the original n-variate input polynomial.

Low dimensional formula reconstruction. We now focus our attention on the `LDR` subroutine. Let f be a homogeneous $(r+1)$-variate polynomial (here $r = O(1)$) which can be computed by a homogeneous ANF formula of depth Δ . Via recursion, it suffices to find homogeneous polynomials f_1, f_2, f_3, f_4 with

$$f = f_1 \cdot f_2 + f_3 \cdot f_4$$

[18] Another technical difficulty we need to handle here is that although the number of variables in this system of polynomial equations is bounded, the number of polynomials themselves is very large - about $2^{d^{O(1)}}$. This is because these polynomials are the appropriate sized minors of a matrix of size $d^{O(1)} \times d^{O(1)}$. We get around this difficulty (in lemma 30 and proposition 31) by observing that these minors all have a small $\mathbb{F}$-basis and this basis can be computed efficiently (in $d^{O(1)}$ time).

[19] The technical reason has to do with the fact that every quadratic form q of rank four can be written as $q = \ell_1 \cdot \ell_2 + \ell_3 \cdot \ell_4$, the ℓ_i's being linear forms, in infinitely many different ways.

such that each f_i can be computed by a homogeneous ANF formula of depth $(\Delta - 1)$. Note that the later condition means in particular that each f_i can be written as

$$f_i = f_{i1} \cdot f_{i2} + f_{i3} \cdot f_{i4}.$$

This subroutine has two steps. In the first step, we proceed as indicated in *Tackling Q2 and Q3* above, and show that if the f_i's satisfy a certain algebraic nondegeneracy condition (which we call formulaic independence), then given $f = (f_1 \cdot f_2 + f_3 \cdot f_4)$, one can efficiently compute a basis for the ideal $\mathfrak{I}(\mathbf{f}) :=< f_1, f_2, f_3, f_4 >$ (lemma 42). Using this basis for $\mathfrak{I}(\mathbf{f})$ and exploiting homogenity, one can easily recover the f_i's upto $\mathbb{F}$-linear combinations (lemma 43). Specifically, we compute polynomials g_1, g_2, g_3, g_4 such that each g_j is a $\mathbb{F}$-linear combination of the f_i's and vice-versa. Our second observation is that if the greatgrandchildren f_{ij}'s satisfy another algebraic nondegeneracy condition (we call it pairwise singular independence) then given the g_j's, one can efficiently compute the f_i's themselves upto scalar multiples (lemma 51). It turns out that recovering the f_i's upto scalar multiples suffices for our purpose. This gives the algorithm. The analysis of this algorithm involves showing that each of the algebraic nondegeneracy conditions are satisfied with high probability at every node of a random homogeneous ANF formula (corollaries 54 and 55). This completes our brief overview.

Organization. We will briefly discuss the significance and limitations of this result in section 1.1 while the remainder of this paper will be devoted to giving a proof of this theorem. In section 2 we will give an overview of the algorithm. Then i section 3 we set up the relevant notation and terminology. For our purpose here, we will need to make several concepts and/or theorems from algebraic geometry more explicit/constructive and we do this in section 4. We then give the low dimensional reconstruction algorithm in section 5. We then use that to give the algorithm of our main theorem in section 6.

3 Preliminaries

3.1 Notations

Index Sets. $[n]$ denotes the set $\{1, 2, \ldots, n\}$ while $[m..n]$ denotes the set $\{m, m+1, \ldots, n\}$. For a finite set S, $\binom{S}{t}$ denotes the set of all subsets of S of size t.

Scalars, Indeterminates and Polynomials. We will use capitalized letters such as X_1, Y_2 etc. to denote formal variables and small letters such as x_1, y_2 etc for elements of a field. Boldfaced letters such as $\mathbf{x}$ (resp. $\mathbf{X}$) shall stand for tuples of field elements (respectively tuples of variables). In a similar fashion we will typically use small letters such as f, g for multivariate polynomials over a field while $\mathbf{f}, \mathbf{g}$ will stand for tuples of polynomials. We will also often think of an m-tuple of polynomials $\mathbf{f} = (f_1, \ldots, f_m) \in (\mathbb{F}[\mathbf{X}])^m$ as an m-dimensional vector over the rational function field $\mathbb{F}(\mathbf{X})$ in the natural way.

Other stuff. "w.h.p." is a shorthand for *with high probability*, specifically with probability $(1 - o(1))$. Let $M(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]^{m \times n}$ be a matrix whose entries are polynomials over the set of variables $\mathbf{X}$ and let $t \geq 1$ be an integer. We will denote by $\text{Minors}(M(\mathbf{X}), t)$ the set of determinants of all $t \times t$ submatrices of $M(\mathbf{X})$. Note that a matrix $M(\mathbf{X})$ has rank less than t if and only if each polynomial in $\text{Minors}(\mathbf{X}, t)$ vanishes identically. For a point $\mathbf{x} \in \mathbb{F}^n$, $\text{rank}(M(\mathbf{x})) < t$ if and only if $\mathbf{x}$ lies on the variety defined by the polynomials in $\text{Minors}(M(\mathbf{X}), t)$. We will sometimes use the shorthand $\partial_i f$ for $\frac{\partial f}{\partial X_i}$.

3.2 Arithmetic formulas, ANF formulas and other representations of polynomials

Arithmetic formulas. Let us fix a variable set $\mathbf{X} = \{X_1, \ldots, X_n\}$, and a field $\mathbb{F}$. An *arithmetic formula* is a rooted binary tree with labels on nodes as follows: the leaves are labeled by elements

from $\mathbf{X} \cup \mathbb{F}$; the internal nodes are labeled by addition ($+$) or product ($\times$). Sometimes, labels on edges are allowed: for the edges feeding into a plus gate, they can be labeled with field constant. Thus it is understood that the plus gate will compute the linear combinations of its two children, using the labels on the edges. An arithmetic formula computes polynomials in the natural way - the polynomial computed at a leaf node is the variable or the field constant that labels it, while the polynomial at a $+$ node (resp. $\times$ node) is the sum (resp. product) of the polynomials computed at the two children. The polynomial outputted at the root is said to be computed by the formula. The *size* of a formula is measured by the number of edges in it.

A generalization of arithmetic formulas is to allow the internal graphs to be an arbitrary directed acyclic graph, rather than a rooted binary tree.

Formulas in alternating normal form. We will be concerned with a special class of formulas as follows: we say that an arithmetic formula ϕ is *in alternating normal form* (in short we say ϕ is an ANF formula) if

1. The underlying tree of ϕ is a complete rooted binary tree (the root node is called the output node). In particular
$$\text{size}(\phi) = 2^{\text{depth}(\phi)+1} - 1,$$
where $\text{size}(\phi)$ is the number of nodes in the tree of ϕ and $\text{depth}(\phi)$ is the maximum distance of a leaf node from the output node of ϕ.

2. The leaves of the tree are labelled with affine forms. i.e. each leaf is labelled with
$$\ell = a_0 + a_1 X_1 + \ldots + a_n X_n,$$
where each $a_i \in \mathbb{F}$ is a scalar.

3. The label of an internal node at distance d from the closest leaf node is $+$ if d is even and $\times$ otherwise. In particular, the root node is a $+$ node, its children are all $\times$ nodes, etc.

Note in particular that there are no labels on the edges going into addition gates in an ANF formula. Thus a $+$ node of an ANF formula computes a simple sum rather than a general $\mathbb{F}$-linear combination of its two inputs. ANF-formulas can be viewed as a canonical form of arithmetic formulas. Converting circuits/formulas into normal forms is standard practice throughout computer science. We now exhibit how to transform any formula into one in ANF form with only a polynomial blow up.

We first note that for ANF formulas, we can assume that each plus gate is a simple sum without any significant loss of generality. Specifically, we have

Fact 4. *Let ϕ be any arithmetic formula computing a polynomial f wherein internal nodes are labelled by $\{+, \times\}$ and arbitrary scalars are allowed on the incoming edges of $+$ nodes so that a $+$ node computes the corresponding $\mathbb{F}$-linear combination of its inputs* [20]*. Then there exists another formula $\hat{\phi}$ computing f of exactly the same size and tree structure as ϕ such that:*

1. *There are no scalars on the edges of $\hat{\phi}$.*

2. *For a leaf node v in ϕ labelled by ℓ, the corresponding leaf node $\hat{v}$ of $\hat{\phi}$ is labelled by $\alpha \cdot \ell$ for some suitable choice of $\alpha \in \mathbb{F}$.*

The proof is an easy induction on the depth of the tree. We next observe that as far as understanding the formula-size complexity of a polynomial is concerned we can focus our attention on formulas in ANF form with only a relatively small loss in the quality of the answer.

[20] Note that subtraction can be done by having the scalar $(-1) \in \mathbb{F}$ on the second incoming edge to a $+$ node

Proposition 5. *Let $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be a polynomial computed by an arithmetic formula ϕ of size s. Then there exists an ANF formula $\hat{\phi}$ of size at most s^4 computing the same polynomial $f(\mathbf{X})$.*

For this, we need the following theorem from Bshouty, Cleve and Eberly (theorem 4 from [BCE91]).

Theorem 6 (Size-depth tradeoffs for algebraic formulas.)**.** *For any arithmetic formula ϕ of size s there exists an arithmetic formula $\hat{\phi}$ of depth $O(\log s)$ and size at most s^2 computing the same polynomial as ϕ.*

Proof of Proposition 5. Using theorem 6 we squashing the original formula to depth $O(\log s)$. We then add dummy nodes (wherein the 'dummy child' of a + gate computes the zero polynomial and the dummy child of a $\times$ gate compute the constant 1) to make the tree a complete binary tree with alternating layers of addition and multiplication gates. This increases the depth by at most a factor of two and hence also the size by at most a quadratic amount. Overall therefore we get an ANF formula of depth at most $O(\log s)$ and size s^4. □

This motivates us to find the smallest ANF formula computing a given polynomial. The rest of this paper is devoted to showing that *random ANF formulas* can be reconstructed efficiently. We first make this notion precise.

Random ANF Formulas. Let $\mathbb{F}$ be a field and $S \subseteq \mathbb{F}$ be a subset of $\mathbb{F}$. Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be an n-tuple of formal variables. We will say that an ANF formula ϕ is a $(\mathbf{X}, \Delta, S)$-ANF formula if it has depth 2Δ and for each linear form labelling

$$\ell = a_0 + a_1 X_1 + a_2 X_2 + \ldots + a_n X_n$$

a leaf node of ϕ, each coefficient a_i is in S. In what follows S will typically be a finite set and we will consider the uniform distribution on the set of $(\mathbf{X}, \Delta, S)$-ANF formulas. Abusing terminology, we will sometimes say that a random (ANF) formula has some property if at least $(1 - o_{|S|}(1))$ fraction of $(\mathbf{X}, \Delta, S)$-ANF formulas have that property.

What does it mean to reconstruct an ANF formula? For a node v of an $(\mathbf{X}, \Delta)$-ANF formula ϕ we denote by $f_v(\mathbf{x})$ the polynomial computed at v and by $\mathbf{f}_v$ the 4-tuple of polynomials computed at the four grandchildren of v. In order to describe the algorithm, we need to define polynomials which in some sense are universal for formulas of product-depth Δ. For each $\Delta \geq 0$, F_Δ will be a 4^Δ-variate polynomial defined recursively as follows.

$$F_0 := X_1$$

Subsequently, $F_{\Delta+1}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4)$ is defined as

$$F_{\Delta+1}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4) := F_\Delta(\mathbf{X}_1) \cdot F_\Delta(\mathbf{X}_2) + F_\Delta(\mathbf{X}_3) \cdot F_\Delta(\mathbf{X}_4),$$

where for $i \in [4]$, we have

$$\mathbf{X}_i := (X_{i1}, X_{i2}, \ldots, X_{i4^\Delta}).$$

Note that F_Δ is a 4^Δ-variate polynomial of degree 2^Δ. Note also that each F_Δ is computed by an ANF formula of size 4^Δ.

Fact 7. *A polynomial $f(\mathbf{X})$ can be computed by a $(\mathbf{X}, \Delta)$-ANF formula of iff there exist affine forms $\ell_1, \ell_2, \ldots, \ell_{4^\Delta}$ such that*

$$f(\mathbf{X}) = F_\Delta(\ell_1, \ell_2, \ldots, \ell_{4^\Delta}).$$

This establishes a one-one correspondence between $(\mathbf{X}, \Delta)$-ANF formulas and 4^Δ-sized tuples of affine forms. Because of this correspondence our ANF formula reconstruction problem can equivalently be

stated as follows: given (blackbox access to) a polynomial f such that there exist $m = 4^{\Delta}$ affine forms $\ell_1, \ldots, \ell_m$ with

$$f = F_{\Delta}(\ell_1, \ell_2, \ldots, \ell_m), \tag{5}$$

find the ℓ_i's. In particular the output of our reconstruction algorithm will simply be a (4^{Δ})-tuple of affine forms satisfying (5).

Representing polynomials in algorithms. Besides arithmetic formulas and arithmetic circuits, another common representation of a polynomial in $\mathbb{F}[X_1, \ldots, X_n]$ is to list the coefficients of the monomials in it. If there is a degree bound d, then the number of monomials in bounded by $\binom{d+n-1}{n-1} = (d+n)^{O(n)}$. Thus this representation is only allowed in the algorithm when n is a constant.

3.3 Preliminaries for polynomials

In this section we collect some standard concepts, and some useful results about polynomials.

DeMillo-Lipton-Schwartz-Zippel lemma.

Lemma 8. *Let $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be an n-variate polynomial of degree d and $S \subseteq \mathbb{F}$ be any subset. If $f(\mathbf{X}) \neq 0$ then $f(\mathbf{x}) = 0$ for at most $\frac{d}{|S|}$ fraction of points $\mathbf{x} \in S^n \subseteq \mathbb{F}^n$.*

$\mathbb{F}$-irreducibility and absolute irreducibility. A polynomial $f(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ is said to be $\mathbb{F}$-reducible if there exist nonconstant polynomials $g(\mathbf{X}), h(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ such that $f(\mathbf{X}) = g(\mathbf{X}) \cdot h(\mathbf{X})$. Otherwise f is said to be $\mathbb{F}$-irreducible. If $f(\mathbf{X})$ is $\bar{\mathbb{F}}$-irreducible then it is said to be absolutely irreducible (here $\bar{\mathbb{F}}$ is the algebraic closure of $\mathbb{F}$).

Homogeneous Components. Recall that a polynomial $f(X_1, \ldots, X_n)$ is said to be *homogeneous* of degree d if every monomial with a nonzero coefficient is of degree d. By collecting together all monomials of the same degree it can be seen that any polynomial $f \in \mathbb{F}[\mathbf{X}]$ of degree d can be uniquely written as

$$f = f^{[d]} + f^{[d-1]} + \ldots + f^{[0]},$$

where each $f^{[i]}$ is homogeneous of degree i. We call $f^{[i]}$ the *homogeneous component of degree i* of f.

Affine forms and linear forms. An *affine form* is a polynomial of degree at most one. A *linear form* is a *homogeneous* polynomial of degree one.

Substitution maps. Let $S \subset [n]$ be a subset of indices of the variables. We shall denote by $\sigma_S : \mathbb{F}[\mathbf{X}] \mapsto \mathbb{F}[\mathbf{X}]$ the ring homomorphism induced by the substitution map

$$\sigma_S(X_i) := \begin{cases} X_i & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

In other words, σ_S preserves the variables in S and "kills" all the other variables by setting them to zero. Occasionally we will be concerned with general linear substitutions which we now define. Let $A = ((a_{ij}))_{n \times n} \in \mathbb{F}^{n \times n}$ be a linear transformation (not necessarily invertible). We will denote by $\sigma_A : \mathbb{F}[\mathbf{X}] \mapsto \mathbb{F}[\mathbf{X}]$ the homorphism given by

$$\begin{aligned} \sigma_A(f(\mathbf{X})) &= f(A \cdot \mathbf{X}) \\ &= f(a_{11}X_1 + \ldots + a_{1n}X_n, \ldots, a_{n1}X_1 + \ldots + a_{nn}X_n) \end{aligned}$$

Let $U \subset \mathbb{F}^n$ be a subspace of dimension t spanned by vectors $\mathbf{a}_1, \ldots, \mathbf{a}_t \in \mathbb{F}^n$. The *restriction of f to the subspace U with respect to the ordered basis* $(\mathbf{a}_1, \ldots, \mathbf{a}_t)$ is defined to be the polynomial

$$g(X_1, \ldots, X_t) \stackrel{\text{def}}{=} f(\mathbf{a}_1 X_1 + \mathbf{a}_2 X_2 + \ldots + \mathbf{a}_t X_t).$$

Stated differently, $g(\mathbf{X}) \stackrel{\text{def}}{=} f(A \cdot \mathbf{X})$, where A is the $n \times n$ matrix whose first t columns consist of the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_t$ respectively and the last $(n-t)$ columns are zero.

$\mathbb{F}$-linear dependence. A very useful notion will be the notion of $\mathbb{F}$-*linear dependencies among polynomials.* We now define this notion.

Definition 9. *Let* $\mathbf{f}(\mathbf{X}) \stackrel{\text{def}}{=} (f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_m(\mathbf{X})) \in (\mathbb{F}[\mathbf{X}])^m$ *be an* m*-tuple of polynomials. The set of* $\mathbb{F}$-linear dependencies *in* $\mathbf{f}$*, denoted* $\mathbf{f}^{\perp}$*, is the set of all vectors* $\mathbf{a} \in \mathbb{F}^m$ *whose inner product with* $\mathbf{f}$ *is the zero polynomial, i.e.,*

$$\mathbf{f}^{\perp} \stackrel{\text{def}}{=} \left\{ (\alpha_1, \ldots, \alpha_m) \in \mathbb{F}^m : \alpha_1 f_1(\mathbf{X}) + \ldots + \alpha_m f_m(\mathbf{X}) = 0 \right\}$$

If $\mathbf{f}^{\perp}$ *contains a nonzero vector, then the* f_i*'s are said to be* $\mathbb{F}$*-linearly dependent else they are* $\mathbb{F}$*-linearly independent.*

The following lemma from [Kay11] is an extension of the DeMillo-Lipton-Scwarz-Zippel lemma and gives an effective criterion for $\mathbb{F}$-linear dependence.

Lemma 10. *Let* $\mathbf{f} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ *be an* m*-tuple of* n*-variate polynomials of degree at most* d *each. Let* $T \subseteq \mathbb{F}$ *be a set. Let*

$$\mathcal{P} := \{\mathbf{a}_i \ : \ i \in [m]\} \subset T^n$$

be a set of m *points in* $\mathbb{F}^n$*. Consider the* $m \times m$ *matrix*

$$M := (f_j(\mathbf{a}_i))_{i,j \in [m]} .$$

With probability at least $(1 - \frac{dm}{|T|})$ *over a random choice of* $\mathcal{P} \in (T^n)^m$*, the nullspace of* M *consists precisely of all the vectors* $(\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{F}^m$ *such that*

$$\sum_{i \in [m]} \alpha_i f_i(\mathbf{x}) = 0.$$

We will also be interested in the dimension of $\mathbf{f}(\mathbf{X}) = (f_1(\mathbf{X}), \ldots, f_m(\mathbf{X}))$ over $\mathbb{F}$. Namely, suppose $\deg(f_i) \leq d$, then f_i's can be viewed as vectors over $\mathbb{F}^{\binom{n+d-1}{n-1}}$, indexed by the monomials in $\mathbf{X}$ with degree $\leq d$. Thus it is natural to consider the dimension of $\mathbf{f}$, denoted as $\dim(\mathbf{f})$. Note that $\mathbf{f}$ is $\mathbb{F}$-linearly independent if and only if $\dim(\mathbf{f}) = m$.

Algebraic dependence. The notion of algebraic dependence between a set of polynomials is defined as follows.

Definition 11 (Algebraic Dependence.)**.** *Let* $\mathbf{f} = (f_1, \ldots, f_m)$ *be an* m*-tuple of polynomials where each* $f_i \in \mathbb{F}[\mathbf{X}]$*. A nonzero polynomial* $A(Z_1, \ldots, Z_m) \in \mathbb{F}[Z_1, \ldots, Z_m]$ *is said to be an* $\mathbf{f}$-annihilating polynomial *if* $A(f_1, \ldots, f_m) = 0$*. The polynomials* $f_1, \ldots, f_m$ *are said to be* algebraically dependent *if there exists an* $\mathbf{f}$*-annihilating polynomial.*

Let $\mathbf{f} = (f_1, \ldots, f_m)$ be an m-tuple of n-variate polynomials. $\mathbb{F}$-linear dependence among the f_i's is a much stronger relationship than algebraic dependence.

Fact 12. *If* $f_1, \ldots, f_m$ *are* $\mathbb{F}$*-linearly dependent then they are algebraically dependent as well.*

Let $J(\mathbf{f}, \mathbf{X})$ be the Jacobian of $\mathbf{f}$ which is defined as the following matrix of partial derivatives.

$$J(\mathbf{f}, \mathbf{X}) \stackrel{\text{def}}{=} \left(\left(\frac{\partial f_i}{\partial X_j} \right) \right)_{m \times n} .$$

This matrix is known as the Jacobian of the set of polynomials in $\mathbf{f}$. The following is a classical theorem (cf. Ehrenborg and Rota [ER93] for a proof).

Theorem 13 (The Jacobian Criterion for algebraic independence.)**.** *Let $f_1, \ldots, f_m \in \mathbb{F}[\mathbf{X}]$ be polynomials over $\mathbb{F}$. If the f_i's are algebraically dependent then the Jacobian matrix, $J(\mathbf{f}, \mathbf{X})$ has rank less than m over $\mathbb{F}(\mathbf{X})$. Moreover if the field $\mathbb{F}$ has characteristic zero then the converse holds true as well, i.e. if the Jacobian matrix, $J(\mathbf{f}, \mathbf{X})$, matrix has rank less than m then the f_i's are algebraically dependent.*

3.4 A subgroup lattice of general linear groups

Group of invertible matrices and its actions. Let $\mathrm{GL}(m, \mathbb{F})$ (simply $\mathrm{GL}(m)$ in short) denote the group of $m \times m$ invertible matrices over $\mathbb{F}$. There is a natural action of $\mathrm{GL}(m)$ on m-tuples of polynomials as follows. For an $\mathbf{f} \in \mathbb{F}(\mathbf{X})^m$ and a matrix $A = ((a_{ij}))_{m \times m} \in \mathrm{GL}(m, \mathbb{F})$ we denote by $A \cdot \mathbf{f} \in \mathbb{F}(\mathbf{X})^m$ the vector

$$\begin{pmatrix} a_{11}f_1 + a_{12}f_2 + \ldots + a_{1m}f_m \\ a_{21}f_1 + a_{22}f_2 + \ldots + a_{2m}f_m \\ \vdots \\ a_{m1}f_1 + a_{m2}f_2 + \ldots + a_{mm}f_m \end{pmatrix} \in \mathbb{F}(\mathbf{X})^m$$

Definition 14. *Let G be a subgroup of $\mathrm{GL}(m, \mathbb{F})$. We will say that two m-tuples $\mathbf{f}, \mathbf{g} \in \mathbb{F}(\mathbf{X})^m$ are* G-equivalent *iff there exists an $A \in G$ such that*

$$\mathbf{g} = A \cdot \mathbf{f}.$$

Note that the fact that G is a subgroup of $\mathrm{GL}(m, \mathbb{F})$ means that 'G-equivalence' is indeed an equivalence relation on m-tuples of polynomials (easy verification). The relevance of this definition to us arises in the following manner. Our algorithm here determines tuples of polynomials computed at some set of internal nodes of the formula but these tuples are determined upto G-equivalence for various choices of G being a subgroup of $\mathrm{GL}(m, \mathbb{F})$. We first introduce subgroups of the symmetric group and then define the subgroups of $\mathrm{GL}(m, \mathbb{F})$ of significance to us. The symmetric group S_m is the group (under composition) of all bijective maps $\pi : [m] \mapsto [m]$. Every element π of S_m corresponds to an invertible linear transformation $A(\pi) \in \mathrm{GL}(m, \mathbb{F})$ in the natural way:

$$A(\pi) \cdot e_i = e_{\pi(i)} \quad \text{where } e_1, e_2, \ldots, e_m \text{ is a basis of } \mathbb{F}^m.$$

For a subgroup G of S_m, abusing notation, we will denote by $G(m, \mathbb{F})$ the corresponding subgroup of $\mathrm{GL}(m, \mathbb{F})$ under the above identification. In particular, $\mathrm{S}(m, \mathbb{F})$ will denote the group of $m \times m$ permutation matrices over $\mathbb{F}$.

Definition 15. *Let $m = 4^{\Delta}$. We define some subgroups of $\mathrm{GL}(m, \mathbb{F})$ as follows.*

(1) $\mathrm{D}(m, \mathbb{F})$ *denotes the group of invertible diagonal matrices. In other words* $\mathrm{D}(m, \mathbb{F})$ *consists of matrices of the form*

$$\begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{pmatrix}, \quad \lambda_i \in \mathbb{F} \setminus \{0\} \quad \forall i \in [n]$$

Structurally, $\mathrm{D}(m, \mathbb{F})$ *is isomorphic to $(\mathbb{F}^*)^m$, the m-wise direct product of $\mathbb{F}^*$.*

(2) $\mathrm{S}(m, \mathbb{F})$ *denotes the group of $m \times m$ permutation matrices over $\mathbb{F}$. It is isomorphic to the symmetric group S_m.*

(3) $\mathrm{DS}(m,\mathbb{F})$ *denotes the subgroup of* $\mathrm{GL}(m,\mathbb{F})$ *generated by* $\mathrm{D}(m,\mathbb{F})$ *and* $\mathrm{S}(m,\mathbb{F})$*. Structurally, it is isomorphic to the semidirect product of* $(\mathbb{F}^*)^m$ *with* S_m*.*

(4) $\mathrm{TR}(m,\mathbb{F})$ *denotes the automorphisms of a rooted complete binary tree of depth* $(\log m) = 2\Delta$*. It is defined recursively as follows.*

- *For* $m = 1$*,* $\mathrm{TR}(1)$ *consists only of the identity matrix.*
- *For* $m > 1$*,* $\mathrm{TR}(m)$ *is the subgroup generated by matrices of the form*

$$\begin{pmatrix} A & \mathbf{0} \\ \mathbf{0} & B \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} \mathbf{0} & \mathbf{1}_{m/2} \\ \mathbf{1}_{m/2} & \mathbf{0} \end{pmatrix}$$

where $A, B \in \mathrm{TR}(m/2,\mathbb{F})$ *and* $\mathbf{1}_{m/2}$ *is the identity matrix in* $\mathrm{GL}(m/2,\mathbb{F})$*.*

(5) $\mathrm{TS}(m,\mathbb{F})$ *denotes the subgroup genrated by* $\mathrm{TR}(m)$ *and matrices of the form*

$$\begin{pmatrix} \alpha\cdot\mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \alpha^{-1}\cdot\mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \beta\cdot\mathbf{1}_{m/4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \beta^{-1}\cdot\mathbf{1}_{m/4} \end{pmatrix}$$

where $\mathbf{1}_{m/4}$ *is the identity matrix in* $\mathrm{GL}(m/4,\mathbb{F})$*.*

(6) $\mathrm{OG}(4,\mathbb{F})$ *denotes the group of invertible* 4×4 *matrices* A *such that*

$$A^T \cdot \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$\mathrm{OG}(4)$ *is the group of symmetries of the quadratic form*

$$q(\mathbf{X}) = X_1\cdot X_2 + X_3\cdot X_4, \ \textit{i.e.}\ q(A\cdot\mathbf{X}) = q(\mathbf{X}) \ \ \textit{iff}\ A\in\mathrm{OG}(4).$$

(7) $\mathrm{FA}(m,\mathbb{F})$ *is defined recursively for* m *being a power of* 4 *as follows.*

- $$\mathrm{FA}(4,\mathbb{F}) \stackrel{\text{def}}{=} \mathrm{OG}(4,\mathbb{F}).$$
- $\mathrm{FA}(m,\mathbb{F})$ *is generated by matrices of the form*

$$\begin{pmatrix} \alpha\cdot A_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \alpha^{-1}\cdot A_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \beta\cdot A_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \beta^{-1}\cdot A_4 \end{pmatrix}, \begin{pmatrix} \mathbf{0} & \mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} \\ \mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_{m/4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_{m/4} \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_{m/4} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_{m/4} & \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{0} & \mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} \\ \mathbf{1}_{m/4} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}_{m/4} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_{m/4} & \mathbf{0} \end{pmatrix}$$

where $\alpha,\beta\in\mathbb{F}^*, A_1, A_2, A_3, A_4 \in \mathrm{FA}(m/4,\mathbb{F})$ *and* $\mathbf{1}_{m/4}$ *is the identity of* $\mathrm{GL}(m/4,\mathbb{F})$*.*

The relationships among these subgroups is given in the form of a lattice diagram in figure 3.4.

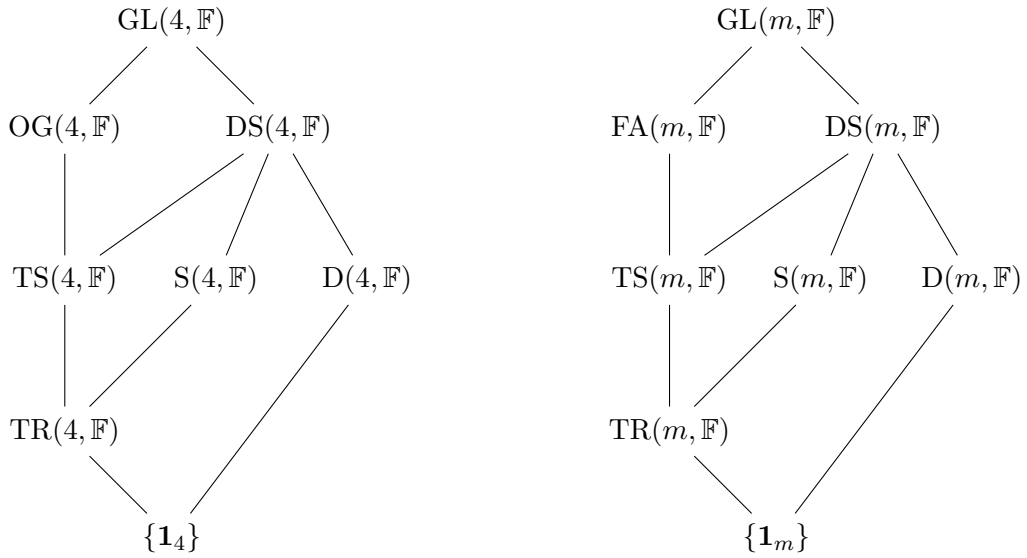

Figure 1: A Lattice of Subgroups of $\mathrm{GL}(4,\mathbb{F}), \mathrm{GL}(m,\mathbb{F})$.

3.5 Summary of concepts from algebraic geometry

Algebraic sets, varieties and ideals. Let $\mathbb{F}$ be a field, $\mathbf{X} = (X_0, X_1, \ldots, X_r)$ be a tuple of $r+1$ indeterminates. $\mathbb{F}[\mathbf{X}]$ denotes the ring of polynomials in $r+1$ variables over $\mathbb{F}$. An *algebraic set* is the set of common zeroes of a system of polynomial equations

$$f_1(\mathbf{X}) = f_2(\mathbf{X}) = \ldots = f_m(\mathbf{X}) = 0.$$

If all the f_i's are homogeneous then such a system also corresponds to a *projective (algebraic) set* wherein two points $\mathbf{x}, \mathbf{y} \in \mathbb{F}^{r+1}$ are considered to be equivalent, denoted $\mathbf{x} \sim \mathbf{y}$, if one is a nonzero scalar multiple of the other. $\mathbb{P}^r(\mathbb{F})$ (simply $\mathbb{P}^r$ for short), called the projective space of dimension r, is the set of all points in $\left(\mathbb{F}^{r+1} \setminus \{\mathbf{0}\}\right)$ modulo this equivalence relation $\sim$. Unless mentioned otherwise, we will always be dealing with projective algebraic sets over an algebraically closed field $\mathbb{F}$. When the algebraic set cannot be written as the union of smaller sets, we will call it an *irreducible variety.*[21]

We will denote by $\mathtt{V}(\mathbf{f})$ the algebraic set defined by the tuple of polynomials $\mathbf{f} = (f_1, \ldots, f_m)$ and by $\mathfrak{I}(\mathbf{f})$ the ideal $\langle f_1, f_2, \ldots, f_m \rangle \subseteq \mathbb{F}[\mathbf{X}]$ generated by these polynomials. When the f_i's are homogeneous polynomials, the ideal $\mathfrak{I}(\mathbf{f})$ is called a *homogeneous ideal* and has the property that for each $g \in \mathfrak{I}(\mathbf{f})$, all the homogeneous components of g also lie in the ideal $\mathfrak{I}(\mathbf{f})$.

Fact 16 (Theorem 2, Page 380 in [CLO07])**.** *An ideal is homogenous if and only if it can be generated by a set of homogeneous polynomials.*

On dimension of algebraic sets. We now examine the dimension of the algebraic set $\mathtt{V}$. We shall use the following definition of dimension from the text by Harris (definition 11.2 of [Har92]). This definition is one among several equivalent definitions of the dimension of an algebraic set.

Definition 17 ([Har92], Definition 11.2)**.** *The dimension of* $\mathtt{V} \subseteq \mathbb{P}^r$*, denoted* $\dim(\mathtt{V})$*, is the smallest integer* k *such that a general* [22] *subspace* $\Lambda \subseteq \mathbb{P}^r$ *of dimension* $(r-1-k)$ *is disjoint from* $\mathtt{V}$*.*

[21] By a *variety* we will simply mean an algebraic set, i.e. the set of zeroes of a system of polynomial equations. Note that for us a variety can be reducible - i.e. it can be expressed as the union of smaller algebraic sets. Note that some authors use the term "variety" to refer to an irreducible variety and hence to avoid confusion we will try to avoid the use of this term.

[22] It means that the subspaces of dimension r satisfying this property form a Zariski dense subset of the Grassmanian variety $G(r,n)$.

The following definition may serve as the definition of dimension for projective sets also.

Proposition 18 ([Har92], Proposition 11.4)**.** *If* $\mathtt{V} \subseteq \mathbb{P}^r$ *is of dimension* k*, then every linear subspace with dimension* $\geq r-k$ *intersects* $\mathtt{V}$ *nontrivially.*

That is, consider using linear subspaces to intersect a projective set of dimension k with dimension i decreasing from r to 0. It will be noted, the situation changes from "always intersecting" to "hardly intersecting" when i goes from $r-k$ to $r-k-1$. This distinction will be exploited in Lemma 24. This behavior can be understood easily when one visualizes using linear subspaces (viewed as in the projective space).

We collect certain well-known facts about dimension of projective algebraic sets.

Fact 19. *Let* $\mathtt{V}, \mathtt{W}$ *be algebraic sets in* $\mathbb{P}^r$*.*

1. *If* $\mathtt{V} \subseteq \mathtt{W}$ *then* $\dim(\mathtt{V}) \leq \dim(\mathtt{W})$*.*
2. $\dim(\mathtt{V} \cup \mathtt{W}) = \max\{\dim(\mathtt{V}), \dim(\mathtt{W})\}$*.*
3. $\dim(\mathtt{V} \cap \mathtt{W}) \leq \min\{\dim(\mathtt{V}), \dim(\mathtt{W})\}$*.*
4. $\dim(\mathtt{V}) = 0$ *if and only if it consists of a finite number of points.*
5. *If* $\mathtt{V}$ *is defined by homogeneous polynomials* $f_1, \ldots, f_k$ *and* $\mathtt{W}$ *is defind by homogeneous polynomials* $g_1, \ldots, g_\ell$ *and if the* f_i*'s and* g_j*'s are pairwise variable-disjoint (i.e. for every* i *and* j *the set of variables in* f_i *is disjoint from the set of variables in* g_j*) then*

$$\operatorname{codim}(\mathtt{V} \cap \mathtt{W}) = \operatorname{codim}(\mathtt{V}) + \operatorname{codim}(\mathtt{W})$$

   or equivalently,

$$\dim(\mathtt{V} \cap \mathtt{W}) = \dim(\mathtt{V}) + \dim(\mathtt{W}) - r.$$

6. (Corollary of Theorem 3, Page 469 in [CLO07]) *If* $\mathtt{V} \subseteq \mathbb{P}^r$ *can be defined by* m *homogeneous polynomials, then* $\dim(\mathtt{V}) \geq r-m$*.*
7. (Irreducible decomposition of an algebraic set) *An algebraic set* $\mathtt{V} \subset \mathbb{P}^r$ *can be decomposed into a finite union of algebraic sets* $\mathtt{W}_1, \mathtt{W}_2, \ldots, \mathtt{W}_u$ *such that*

   (a) Every $\mathtt{W}_i$ *is irreducible.*

   (b) $W_i \not\subseteq W_j$ *for* $i \neq j$*,* $i, j \in [u]$*.*

   The algebraic sets $\mathtt{W}_i$ *(called the irreducible components of* $\mathtt{V}$*) are uniquely determined.*
8. (Equidimensional decomposition of an algebraic set) *For each* $0 \leq i \leq r$*, let* $\mathtt{V}_i$ *be the union of all irreducible components of* $\mathtt{V}$ *(possibly none) of dimension* i*. Then* $\mathtt{V} = \bigcup_{i=0}^{r} \mathtt{V}_i$*. This is called the* equidimensional decomposition of $\mathtt{V}$*.*

On singularity of algebraic sets. We have seen that the Jacobian matrix is related to algebraic independence of a set of polynomials. In algebraic geometry it has to do with the singular points. Recall that $J(\mathbf{f}, \mathbf{X}) \in (\mathbb{F}[\mathbf{X}])^{m \times (r+1)}$ is defined as the $m \times (r+1)$ matrix whose (i,j)-th entry is $\frac{\partial f_i}{\partial X_j}$. Thus

$$J(\mathbf{f}, \mathbf{X}) = \begin{pmatrix} \frac{\partial f_1}{\partial X_0} & \frac{\partial f_1}{\partial X_1} & \cdots & \frac{\partial f_1}{\partial X_r} \\ \frac{\partial f_2}{\partial X_0} & \frac{\partial f_2}{\partial X_1} & \cdots & \frac{\partial f_2}{\partial X_r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial X_0} & \frac{\partial f_m}{\partial X_1} & \cdots & \frac{\partial f_m}{\partial X_r} \end{pmatrix}$$

The significance of $J(\mathbf{f}, \mathbf{X})$ is as follows. If we view $J(\mathbf{f}, \mathbf{X})$ as an $m \times (r+1)$ matrix over the function field $\mathbb{F}(\mathbf{X})$ its rank denotes the dimension of the image of the map $\Psi : \mathbb{F}^{r+1} \mapsto \mathbb{F}^m, \mathbf{x} \mapsto (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$. The dimension of the nullspace of $J(\mathbf{f}, \mathbf{X})$ is then the dimension of the preimage of *a generic point in the image of the map* Ψ. On the other hand, the dimension of $\mathtt{V}(\mathbf{f})$ is the dimension of $\Psi^{-1}(\mathbf{0})$. For a point $\mathbf{x} \in \mathtt{V}$, it is always the case that $\dim(\text{NullSpace}(J(\mathbf{f}, \mathbf{x}))) \geq 1 + \dim_{\mathbf{x}}(\mathtt{V})$ holds [23]. We say that $\mathbf{x} \in \mathtt{V}$ is a *smooth point of* $\mathtt{V}$ iff

$$\dim(\text{NullSpace}(J(\mathbf{f}, \mathbf{x}))) = 1 + \dim_{\mathbf{x}}(\mathtt{V}).$$

Otherwise $\mathbf{x}$ is said to be a *singular point of* $\mathtt{V}$. The *singularity* of $\mathtt{V}(\mathbf{f})$ is defined as the set of all singular points in in.

A case of particular interest will be the singular points of a hypersurface - i.e. a variety defined by a single polynomial $f(\mathbf{X})$. For a polynomial $f(\mathbf{X})$, we will denote by $\text{Sing}(f)$, the variety consisting of singularities of $\mathtt{V}(f)$, i.e. the set of points $\mathbf{x} \in \mathbb{F}^n$ satisfying

$$f(\mathbf{x}) = \frac{\partial f}{\partial X_1}(\mathbf{x}) = \ldots = \frac{\partial f}{\partial X_n}(\mathbf{x}) = 0.$$

When f is homogeneous,

$$\text{Sing}(f) \equiv \mathtt{V}(\frac{\partial f}{\partial X_1}, \frac{\partial f}{\partial X_2}, \ldots, \frac{\partial f}{\partial X_n}).$$

Namely, we can drop the condition that $\mathbf{x} \in \mathtt{V}$ in the definition of a singular point. This is because for a homogeneous polynomial f, the vanishing of all its partial derivatives ensures the vanishing of f, by the following well-known identity:

$$f(\mathbf{X}) = \frac{1}{\deg(f)} \cdot \left(\sum_{i=n}^{n} X_i \cdot \frac{\partial f}{\partial X_i} \right).$$

Commutative algebraic formulation of algebraic geometry. While the geometric formulation above has the advantage of being quick and intuitive, the algebraic formulation is closer to reality – in the algorithm, we need to deal with a few polynomials viewed as a generating set of an ideal.

Let $\mathfrak{I}$ be an ideal in $\mathbb{F}[\mathbf{X}]$. The *radical* of $\mathfrak{I}$, denoted as $\sqrt{\mathfrak{I}}$ is $\{f \in \mathbb{F}[\mathbf{X}] \mid \exists n \in \mathbb{N}, f^n \in \mathfrak{I}\}$. We say $\mathfrak{I}$ is *radical* if $\sqrt{\mathfrak{I}} = \mathfrak{I}$. *Hilbert's Nullstellensatz* shows the correspondence between algebraic sets and radical ideals when $\mathbb{F}$ is algebraically closed. Formally, it states that if $\mathbb{F}$ is algebraically closed, then $\mathfrak{I}(\mathtt{V}(\mathfrak{I})) = \sqrt{\mathfrak{I}}$. As we will see, the fact that algebraic sets correspond to radical ideals will cause certain complication to the algebraic formulation.

Recall that for an algebraic set, there can be irreducible decomposition and equidimensional decomposition for it. We will present the algebraic correspondents of the two decompositions.

Let $\mathfrak{I}$ be an ideal in $\mathbb{F}[\mathbf{X}]$. $\mathfrak{I}$ is *primary* if for $f, g \in \mathbb{F}[\mathbf{X}]$, $fg \in \mathfrak{I}$ implies that either f or g^m ($m \in \mathbb{N}$) is in $\mathfrak{I}$. The *primary decomposition* of $\mathfrak{I}$ is to express $\mathfrak{I}$ as the intersection of primary ideals like $\mathfrak{I} = \bigcap_{i=1}^{\ell} \mathfrak{J}_i$, where $\mathfrak{J}_i$'s are primary. This decomposition is *irredundant* if there are no inclusion relations between any two of $\sqrt{\mathfrak{J}_i}$'s. The *Lasker-Noether theorem* asserts that for a polynomial ideal, a irredundant primary decomposition exists. In the following, a primary decomposition would be irredundant unless stated otherwise explicitly. In fact, the primary decomposition of $\mathfrak{I}$ corresponds to the irreducible decomposition of $\mathtt{V}(\mathfrak{I})$ in an exact sense: if $\mathfrak{I}$ is primary, then $\sqrt{\mathfrak{I}}$ satisfies that if $fg \in \mathfrak{I}$ then either f or g is in $\mathfrak{I}$. An ideal satisfying this property is *prime*. Thus in the primary decomposition $\mathfrak{I} = \bigcap_{i=1}^{\ell} \mathfrak{J}_i$, $\sqrt{\mathfrak{J}_i}$ are called the *associated prime ideals* of $\mathfrak{I}$. For an algebraic set $\mathtt{V}$, $\mathtt{V}$ is irreducible, if and only if $\mathfrak{I}(\mathtt{V})$ is a prime ideal. Furthermore, the irreducible decomposition of $\mathtt{V}(\mathfrak{I})$ is exactly $\bigcup_i \mathtt{V}(\mathfrak{J}_i)$.

[23] The '+1' here is because we are looking at the dimension in projective space rather than affine space

To explain the equidimensional decomposition of an ideal $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$, we start with a discussion on the definition of dimension for a polynomial ideal. In general, for a Noether ring R, the *Krull dimension* of R is the largest number d such that there exists a strictly increasing chain of prime ideals $P_0 \subset P_1 \subset \cdots \subset P_d$ in R. (Note that R is not a prime ideal.) For polynomial rings, the Krull dimension and Definition 17 coincide. We can understand the dimension of $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$ as that of $\mathtt{V}(\mathfrak{I})$.

Now it is straightforward to see what the equidimensional decomposition of an ideal $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$ would be: suppose $\mathfrak{I}$ is of dimension d. Then in the primary decomposition $\mathfrak{I} = \bigcap_{i=1}^{\ell} \mathfrak{J}_i$, for each $j \in \{0, 1, \ldots, d\}$, form $\mathfrak{I} = \bigcap_{j=0}^{d} \mathfrak{I}_j$, where $\mathfrak{I}_j$ is the intersection of $\mathfrak{J}_i$'s of dimension j. This is the equidimensional decomposition of $\mathfrak{I}$. The *top dimensional component* is of course $\mathfrak{I}_d$, denoted as $\mathrm{top}(\mathfrak{I})$.

4 Explicit versions of some Algebraic Geometry concepts

In this sections we revisit some standard notions from algebraic geometry with a view towards making these concepts explicit and enough for our purpose. While most of the tools and concepts from algebraic geometry that we use here are already available, some of them are not available in the form that we need here and so in this section we develop these concepts further and make things more explicit. In Section 4.1 we will see how the dimension of a projective algebraic set $\mathtt{V}$ can be captured by an explicit polynomial matrix, where entries are polynomials in the coefficients of the defining polynomials of $\mathtt{V}$. We also mention an algorithm for solving polynomial equations by Lazard [Laz81] (cf. [Laz01]) in this section. In Section 4.2 an algorithmic trick for handling polynomial matrices is presented. Combining results from Section 4.1, we are able to determine the dimension of an algebraic set. In Section 4.3 we exhibit a well-known algorithm to extract the top dimensional component of an ideal, and present an implementation that allows worst-case analysis.

4.1 The resultant system for a set of homogeneous polynomials

In this section, we describe the resultant system for a set of homogeneous polynomials, which will be a major algorithmic tool for us. To illustrate this concept, we feel it helpful to review the classical resultant for two monic univariate polynomials $P(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ and $Q(x) = x^m + b_{m-1}x^{m-1} + \cdots + b_1 x + b_0$ in $\mathbb{F}[x]$. The *resultant* of $P(x)$ and $Q(x)$ $\textsc{Res}(P, Q) = \prod_{(r,s):P(r)=0,Q(s)=0}(r - s)$. Interestingly, $\textsc{Res}(P, Q)$ is equal to the determinant of the $(n + m) \times (n + m)$ *Sylvester matrix* w.r.t. P and Q:

$$\textsc{Syl}(P, Q) = \begin{pmatrix} 1 & a_{n-1} & \ldots & a_0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & a_1 & a_0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & 0 & 1 & a_{n-1} & \ldots & a_0 \\ 1 & b_{m-1} & \ldots & b_0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & b_1 & b_0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & 0 & 1 & b_{m-1} & \ldots & b_0 \end{pmatrix}$$

The equivalence of $\det(\textsc{Syl}(P, Q))$ and $\textsc{Res}(P, Q)$ can be interpreted as follows. Note that $\textsc{Res}(P, Q) = 0$ if and only if P and Q share a common zero. Also note that Sylvester matrix is determined by the defining parameters of P and Q. Thus, the geometric fact (whether having a common zero) is reflected in the algebraic manipulation of the defining coefficients of P and Q (whether determinant of the Sylvester matrix is zero).

We would like to generalize the classical resultant for univariate polynomials, to a resultant system for a set of homogeneous polynomials. A treatment of the resultant system can be found in [Yap00].

The setting. Let $\mathbb{F}$ be an algebraically closed field of characteristic zero. Let $\mathbb{S} \subset \mathbb{F}$ be a set and let $\mathbf{X} = (X_0, X_1, \ldots, X_r)$ be an $(r+1)$-tuple of indeterminates. For an integer $d \geq 0$, we denote by

$$\mathbf{L}_d := \{(i_0, i_1, i_2, \ldots, i_r) \in (\mathbb{Z}_{\geq 0})^{r+1} \ : \ (i_0 + i_1 + i_2 + \ldots + i_r) = d\},$$

the set of all possible indices of monomials of total degree d over the $(r+1)$-tuple $\mathbf{X}$. We denote by $L_d = \binom{r+d}{d}$ the size of $\mathbf{L}_d$. Let $\mathbf{g}(\mathbf{X}) = \{g_1, g_2, \ldots, g_m \mid g_i \in \mathbb{F}[\mathbf{X}]\}$ be *homogeneous* polynomials of degree d over the variable set $\mathbf{X}$. For $j \in [m]$, let

$$g_j(\mathbf{X}) = \sum_{\mathbf{i} \in \mathbf{L}_d} a_{\mathbf{i}j} \mathbf{X}^{\mathbf{i}}.$$

Let $\mathbf{a} = (a_{\mathbf{i}j})_{\mathbf{i} \in \mathbf{L}_d, j \in [m]}$ be the vector of coefficients of the g_j's.

Now we define "universal polynomials" to capture the idea that a set of homogeneous polynomials as above is specified by the coefficients of the monomials. Let $\mathbf{A} = (A_{\mathbf{i}j})_{\mathbf{i} \in \mathbf{L}_d, j \in [m]}$ be a vector of formal variables. Then define the universal polynomial $\mathbf{f}(\mathbf{A}, \mathbf{X}) = \{f_1, \ldots, f_m \mid f_i \in \mathbb{F}[\mathbf{A}, \mathbf{X}]\}$, where

$$f_j(\mathbf{X}) = \sum_{\mathbf{i} \in \mathbf{L}_d} A_{\mathbf{i}j} \mathbf{X}^{\mathbf{i}}.$$

To recover $\mathbf{g}$ from $\mathbf{f}$, it is enough to assign $\mathbf{a} \in \mathbb{F}^{L_d \times m}$ to $\mathbf{A}$. Denote $\mathbf{f}_{\mathbf{a}} = \{f_1(\mathbf{a}, \mathbf{X}), \ldots, f_m(\mathbf{a}, \mathbf{X}) \mid f_i\}$, and it is clear that $\mathbf{f}_{\mathbf{a}} = \mathbf{g}$.

Characterizing whether a projective set is empty or not.

Definition 20. *A resultant system for* $\mathbf{f}$ *is a set of homogeneous polynomials* $\mathbf{p} = \{p_1, \ldots, p_\ell \mid p_i \in \mathbb{F}[\mathbf{A}]\}$*, such that for* $\mathbf{a} \in \mathbb{F}^{L_d \times m}$*,* $\mathtt{V}(\mathbf{f}_{\mathbf{a}}) \neq \emptyset$ *if and only if* $\mathbf{a} \in \mathtt{V}(\mathbf{p})$*.*

Note that from the definition it is not a priori clear that a resultant system for a set of polynomials should exist. For example, we can come up with the same definition of resultant systems for sets of nonhomogeneous polynomials, then it can be exhibited that such a resultant system would not exist. For homogeneous polynomials, the existence of resultant systems is ensured by the main theorem of elimination theory (cf. e.g. [CLO05, Chap. 8]). In [Yap00], a constructive proof of the existence of resultant system is presented in Lecture 11, Section 4. We adapt that proof to prove Lemma 24. To get an explicit bound in Lemma 24, the so-called *effective Nullstellensatz* will be crucial. Recall that Hilbert's Nullstellensatz states that when $\mathbb{F}$ is algebraically closed, a polynomial f vanishes on the zeros of an ideal $\mathfrak{I} \subseteq \mathbb{F}[X_0, X_1, \ldots, X_r]$, if and only if there exists an integer e, such that $f^e \in \mathfrak{I}$. Suppose there is a degree bound d on a set of generators for $\mathfrak{I}$, then the effective Nullstellensatz, or quantitative Nullstellensatz as in [Yap00], establishes bound for the exponent e on d and r. We cite the following bound by Dubé [Dub93].

Theorem 21 ([Dub93], Theorem 7.1)**.** *Situations as above. Let* $M = 13d^{r+1}$*. If* $f \in \sqrt{\mathfrak{I}}$*, then* $f^M \in \mathfrak{I}$*.*

Lemma 22 (Resultant system for projective sets)**.** *Let the notations be as above. Then there exists a matrix* $B(\mathbf{A}) \in (\{0\} \cup \mathbf{A})^{s \times t}$*, such that for* $\mathbf{a} \in \mathbb{F}^{L_d \times m}$*,* $\mathtt{V}(\mathbf{f}_{\mathbf{a}}) \neq \emptyset$ *if and only if* $\operatorname{rank}(B(\mathbf{a})) < s$*. Moreover, for* $d \geq r$*, we have* $s \leq d^{O(r^2)}$ *and* $t \leq m \cdot d^{O(r^2)}$*.*

In particular, this means that the $s \times s$ *minors of* B *is a resultant system for* $\mathbf{f}$*.*

Proof. Fix $M = 13d^{r+1}$ from now. As we are aiming for nontrivial zeros of $\mathtt{V}(\mathfrak{I})$ (that is, we understand $\mathtt{V}(\mathfrak{I})$ as in projective space), then the condition of $\mathtt{V}(\mathfrak{I})$ being empty translates to $\sqrt{\mathfrak{I}} = \langle X_0, X_1, \ldots, X_r \rangle$. By Theorem 21, $\mathtt{V}(\mathfrak{I}) = \emptyset$ if and only if $\langle X_0, X_1, \ldots, X_r \rangle^M \subseteq \mathfrak{I}$. Furthermore, let $N = 1 + (r+1)(M-1)$, and $S = \{\mathbf{X}^{\mathbf{i}} \ : \ \mathbf{i} \in \mathbf{L}_N\}$ be the set of monomials of degree N. Then we see that $\mathtt{V}(\mathfrak{I}) = \emptyset$ if and only if $S \subseteq \mathfrak{I}$.

Now we can present the construction as in [Yap00]. The idea is to express each monomial $\mathbf{X}^{\mathbf{i}} \in S$ as being in the ideal $\mathfrak{I}$. Suppose $\mathbf{X}^{\mathbf{i}} = h_1 \cdot g_1 + h_2 \cdot g_2 + \ldots + h_m \cdot g_m$, where h_i's are homogeneous polynomials of degree $N - d$. Viewing the coefficients of h_i's as variables, this gives a linear equation with $m \cdot L_{N-d}$ variables. For every monomial in $\mathbf{X}$ of degree N, such a linear equation can be formed. So we have L_N linear equations in $m \cdot L_{N-d}$ variables, and these linear equations can be written as $B \cdot V = I$, where B is an $L_N \times (m \cdot L_{N-d})$ matrix, whose entries are coefficients of g_j's. B is the matrix as desired in the statement; it can be defined formally as follows: the rows are indexed by monomials of degree N, and the columns are indexed by monomials in h_i's. Given a $\mathbf{X}^{\mathbf{i}}$, $|\mathbf{i}| = N$ and $(j, \mathbf{X}^{\mathbf{k}})$, $j \in [m]$ and $|\mathbf{k}| = N - d$, if $\mathbf{i} \geq \mathbf{k}$ (namely, at each coordinate $e \in \{0, 1, \ldots, r\}$, $\mathbf{i}_e \geq \mathbf{k}_e$), $B(\mathbf{i}, (j, \mathbf{k}))$ is the coefficient of $\mathbf{X}^{\mathbf{i}-\mathbf{k}}$ in g_j. If $\mathbf{i} < \mathbf{k}$ the entry is 0. V is an $(m \cdot L_{N-d}) \times L_N$ matrix, whose entries represent the solutions to the linear equations. I is an $L_N \times L_N$ identity matrix. Then $S \subseteq \mathfrak{I}$ if and only if each linear equation has a solution, and this happens if and only if $\mathrm{rank}(B) = L_N$. The latter condition can be written as the open set defined by all the $L_N \times L_N$ minors of B, proving the "moreover" part in our statement. In particular, if $m \cdot L_{N-d} < L_N$, from the above discussion, the system of polynomial equations always have a nontrivial common zero. We leave it for the reader to verify that when $m \geq 2$ and when N is large enough, then $m \cdot L_{N-d} \geq L_N$. On the other hand, if $m \cdot L_{N-d} \geq L_N$, it is still possible that every minor is a zero polynomial thus putting no constraints on the $\mathbf{A}$ variables.

Note that the minors are have degree bounded by $L_N = \binom{r+N}{r} = \binom{(r+1)M}{r} = d^{O(r^2)}$. We finally remark that the degree of the classical multivariate resultant of $r+1$ polynomials of degree d is $(r+1)d^r$ (cf. Theorem 3.1 in [CLO05]). □

Characterizing whether a projective set has dimension $\geq k$ or not. Now consider a set of $t+1$ points $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_c$ in $\mathbb{P}^r$. The span of these points, namely the set

$$\{y_0 \cdot \mathbf{v}_0 + y_1 \cdot \mathbf{v}_1 + \ldots + y_c \cdot \mathbf{v}_c \quad : \quad y_0, y_1, \ldots, y_c \in \mathbb{F}\} \subseteq \mathbb{P}^r$$

is a subspace of dimension at most c. It is of dimension equal to c if and only if $\mathbf{v}_0, \mathbf{v}_1, \ldots, \mathbf{v}_c$ are linearly independent. The following proposition indicates how to express the condition $\dim \geq k$ in an algebraic way; it is a consequence of Definition 17 and Proposition 18.

Proposition 23. *Let $k \geq 1$ be an integer and $c = (r - k)$. Then $\mathtt{V}(\mathbf{f})$ has dimension $\geq k$ if and only if for all* linearly independent *$\mathbf{v}_0, \ldots, \mathbf{v}_c \in \mathbb{P}^r$ the projective algebraic set defined by the homogeneous polynomials*

$$\{f_i(Y_0 \cdot \mathbf{v}_0 + \ldots + Y_c \cdot \mathbf{v}_c) \in \mathbb{F}[Y_0, Y_1, \ldots, Y_c] \quad : \quad i \in [m]\} \subset \mathbb{F}[\mathbf{Y}]$$

has a common nontrivial solution for the variables $(Y_0, \ldots, Y_c)$.

In the following, we use $\mathbf{V} = (\mathbf{V}_0, \ldots, \mathbf{V}_c)$ to denote a tuple of $(c+1)(r+1)$ formal variables that are going to be substituted by vectors $\mathbf{v}_0, \ldots, \mathbf{v}_c$ from $\mathbb{P}^r$. Combining this proposition with Lemma 22 we get the following lemma.

Lemma 24 (Resultant system for projective sets with dimension $\geq k$)**.** *Let the notations be as above. Then there exists a matrix $B(\mathbf{V}, \mathbf{A}) \in (\mathbb{Z}[\mathbf{V}, \mathbf{A}])^{s \times t}$, such that for $\mathbf{a} \in \mathbb{F}^{L_d \times m}$, $\dim(\mathtt{V}(\mathbf{f}_\mathbf{a})) \geq k$ if and only if every $s \times s$ minor of $B(\mathbf{V}, \mathbf{a})$ is a zero polynomial in $\mathbb{F}[A]$. Moreover, for $d \geq r$, we have $s \leq d^{O(c^2)}$ and $t \leq m \cdot d^{O(c^2)}$, and the degree of the entries in B is bounded by d.*

Proof. In the universal polynomial $\mathbf{f}$, set $(X_0, \ldots, X_r)$ to $Y_0\mathbf{V}_0 + \ldots + Y_c\mathbf{V}_c$. Let $\mathbf{h} = (h_0, \ldots, h_k)$ be the tuple of polynomials after the replacement; note that $h_i \in \mathbb{F}[\mathbf{V}, \mathbf{A}][\mathbf{Y}]$. As in the proof of Lemma 22, we build a matrix B consisting of coefficients of the $\mathbf{Y}$ variables in h_i's. Let $M = 13d^{c+1}$ and $N = 1 + (c+1)(M-1)$. Then B is of size L_N by mL_{N-d}, with entries from $\mathbb{Z}[\mathbf{V}, \mathbf{A}]$. Note that $\mathrm{Minors}(B, L_N)$ consists of polynomials from $\mathbb{Z}[\mathbf{A}][\mathbf{V}]$.

Now consider a specific tuple of polynomials $\mathbf{f_a}$. Then for a particular assignment $\mathbf{v}$ to $\mathbf{V}$, $h_1|_{\mathbf{v}}, \ldots, h_m|_{\mathbf{v}}$ have a nontrivial common zero if and only if for every $q(\mathbf{V}) \in \text{Minors}(B_{\mathbf{a}}, L_N)$, $q(\mathbf{v}) = 0$. Thus we use Definition 17 and Proposition 18 to distinguish whether $\dim(V) \geq k$.

- If $\dim(V) \geq k$, then for every linearly independent $\mathbf{v} = (\mathbf{v}_0, \ldots, \mathbf{v}_c)$, and every $q(\mathbf{V}) \in \text{Minors}(B, L_N)$ we have $p(\mathbf{v}) = 0$. In particular, this implies that every $q(\mathbf{V}) \in \text{Minors}(B, L_N)$ vanishes on a Zariski-dense subset, thus $q(\mathbf{V}) \equiv 0$.
- If $\dim(V) < k$, then for a general $\mathbf{v} = (\mathbf{v}_0, \ldots, \mathbf{v}_c)$, there exists $q(\mathbf{V}) \in \text{Minors}(B, L_N)$, $q(\mathbf{v}) \neq 0$. This implies the existence of a nonzero polynomial $q(\mathbf{V}) \in \text{Minors}(B, L_N)$.

That is, $\mathtt{V}(\mathbf{f_a})$ has dimension $\geq k$ if and only if every $q(\mathbf{a}, \mathbf{V}) \in \text{Minors}(B, L_N)$, viewed as $\in \mathbb{F}[\mathbf{V}]$, is a zero polynomial. This proves the lemma.

Let us get back to the case of universal polynomials. For every $q(\mathbf{A}, \mathbf{V}) \in \text{Minors}(B, L_N)$, let U be the set of the the coefficients of $\mathbf{V}$ variables, which are polynomials from $\mathbb{F}[\mathbf{A}]$. The above discussion implies that, $\mathbf{f_a}$ has dimension $\geq k$ if and only if $\mathbf{a}$ satisfies the polynomials in U. In other words, U characterizes whether $\dim(\mathbf{f_a}) \geq k$. □

Solving polynomial equations using resultant. For solving polynomials we will be concerned with the *affine* case; namely, polynomials need not be homogeneous.

A well-known application of resultant system is for solving a system of polynomials, cf. e.g. [CLO05]. In this paper we are concerned with solving a system of polynomials defining a 0-dimensional (affine) algebraic set. In this 0-dimensional case, solving a system of polynomial equations has evolved into a mature discipline (cf. Lazard's survey [Laz09]), and is widely implemented (mostly via Gröbner basis). For the sake of worst-case analysis, we refer to a result by Lazard ([Laz81], cf. [Laz01]) 30 years ago. We note that this result builds on the u-resultant technique, and is particularly suited for complexity analysis in our setting.

Theorem 25 (Theorem 8.1, [Laz81], cf. [Laz01])**.** *Given polynomials $f_1, \ldots, f_m \in \mathbb{F}[X_1, \ldots, X_r]$ with degree bounded by d, suppose $\mathtt{V}(f_1, \ldots, f_m)$ is of dimension 0 and consists of N points with multiplicity 1. Let $D = \max(d, 3)$. Then there exists an algorithm solving $(f_1 = 0, \ldots, f_m = 0)$ using: (1) $O(mD^{O(r)})$ field operations; (2) $O(rN^4)$ operations in degree-N extensions of $\mathbb{F}$; (3) solving degree-N univariate polynomial over $\mathbb{F}$.*

In our application the polynomials are homogeneous, defining projective algebraic set of dimension 0. To apply Theorem 25 we can "dehomogenize" the polynomials by setting a variable to 1, and use the theorem. Do this for every variable and collect the solutions afterwards.

4.2 Randomized algorithms for polynomial matrices

A polynomial matrix $B(\mathbf{X})$ is a matrix from $(\mathbb{F}[\mathbf{X}])^{s \times t}$, where $\mathbf{X} = \{X_1, \ldots, X_n\}$. Assume the degrees of the entries in $B(\mathbf{X})$ to be bounded by d. Let $r \leq \min(s, t)$ be an integer. In this section, we would like to determine those $\mathbf{x} \in \mathbb{F}^n$ such that $\text{rank}(B(\mathbf{x})) < r$. Note that $B(\mathbf{x})$ is of rank less than r if and only if all $r \times r$ submatrices of $B(\mathbf{x})$ have determinant zero, in other words $\mathbf{x} \in \mathtt{V}(\text{Minors}(B(\mathbf{X}), r))$. We can do this by computing down all the $m = \binom{s}{r} \cdot \binom{t}{r}$ minors of $B(\mathbf{X})$ and then applying theorem 25. The difficulty here is that m can be very large - for example when $s = t = 2r$ then $m = 2^{O(r)}$ so that a naïve application theorem 25 can lead to a running time which is exponential with respect to r. We want to avoid this exponential dependence on r and obtain a running time in which only the number of variables n appears in the exponent. In our application eventually n will be a constant (actually 4) so that we want a running time which is polynomial in all the other parameters. The idea is to exploit the structure of $\mathbf{f} := \text{Minors}(B(\mathbf{X}), s)$; the property of interest is its dimension over $\mathbb{F}$ (cf. Section 3.1, definition 9).

Proposition 26. *Let* $\mathbf{f}$ *and* $\mathbf{g}$ *be any two tuples of polynomials (not necessarily of the same length). If* $\mathrm{Span}(\mathbf{f}) = \mathrm{Span}(\mathbf{g})$ *then* $\mathtt{V}(\mathbf{f}) = \mathtt{V}(\mathbf{g})$.

Proof. $\mathrm{Span}(\mathbf{g}) = \mathrm{Span}(\mathbf{f})$ means that every polynomial in $\mathbf{g}$ is an $\mathbb{F}$-linear combination of polynomials in $\mathbf{f}$ and vice-versa. In particular, since every polynomial in $\mathbf{f}$ is an $\mathbb{F}$-linear combination of polynomials in $\mathbf{g}$ we have $\mathtt{V}(\mathbf{g}) \subseteq \mathtt{V}(\mathbf{f})$. Similarly since every polynomial in $\mathbf{g}$ is an $\mathbb{F}$-linear combination of polynomials in $\mathbf{f}$ we have $\mathtt{V}(\mathbf{f}) \subseteq \mathtt{V}(\mathbf{g})$. The proposition follows. □

By counting the number of monomials possibly appearing in a $(r \times r)$ minor of $B(\mathbf{X})$, we obtain the following trivial bound for $\dim(\mathrm{Span}(\mathbf{f}))$.

Fact 27. $\dim(\mathrm{Span}(\mathbf{f}) \leq \binom{dr+n+1}{n}$.

We now fix $m = \binom{dr+n+1}{n}$. We will compute an m-tuple of polynomials which has the same $\mathbb{F}$-span as $\mathbf{f}$.

Proposition 28. *For a set of* m *randomly chosen pairs of matrices*

$$\{(A_i, C_i) \in (\mathbb{F}^{r\times s}) \times (\mathbb{F}^{t\times r}) \quad : \quad i \in [m]\},$$

the m*-tuple of polynomials* $\mathbf{g} = (\det(A_i \cdot B(\mathbf{X}) \cdot C_i))_{i\in[m]}$ *satisfies* $\mathrm{Span}(\mathbf{g}) = \mathrm{Span}(\mathbf{f})$ *with probability at least* $1 - \frac{2rm}{|\mathbb{F}|}$.

Proof. For any $(A, C) \in (\mathbb{F}^{r\times s}) \times (\mathbb{F}^{s\times r})$ we have we have $\det(A \cdot B(\mathbf{X}) \cdot C) \in \mathrm{Span}(\mathrm{Minors}(B, r))$ so that $\mathrm{Span}(\mathbf{g}) \subseteq \mathrm{Span}(\mathbf{f})$ holds true *always*. Assume that $f_1(\mathbf{X}), f_2(\mathbf{X}), \ldots, f_k(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ $(k \leq m)$ is a maximal set of $\mathbb{F}$-linearly independent polynomials in $\mathbf{f}$. It suffices to show that $\dim(\mathrm{Span}(\mathbf{g})) = k$ with high probability. Now understand the entries in A_i's and C_i's as formal variables, denoted by the variable set $\mathbf{Y}$ of size $2mst$. Viewing g_i's as polynomials over $\mathbb{F}(\mathbf{Y})[\mathbf{X}]$, whether $\dim(\mathrm{Span}(\mathbf{g})) = k$ is characterized by a set of polynomials (of degree $2kr$) in $\mathbf{Y}$. Thus to show that $\dim(\mathrm{Span}(\mathbf{g})) = k$ w.h.p. it is enough to exhibit a specific assignment to $\mathbf{Y}$ such that $\dim(\mathrm{Span}(\mathbf{g})) = k$ for this specific assignment to $\mathbf{Y}$. This specific assignment can be got, by letting A_i's and the C_i's "choose" the linearly independent minors $f_1, \ldots, f_k$. The probability bound follows via an application of the DeMillo-Lipton-Schwarz-Zippel lemma (lemma 8). □

Combining this proposition with Lemma 24, we get the following proposition determining the dimension of an algebraic set. It follows by applying Proposition 28, to the polynomial matrix $B(\mathbf{V}, \mathbf{a})$ from Lemma 24 for each $k \in \{0, \ldots, r\}$, to check whether the span of minors of $B(\mathbf{V}, \mathbf{a})$ is of dimension 0.

Proposition 29. *Let notations be as in Lemma 24. Then there exists a randomized algorithm that computes* $\dim(\mathtt{V}(\mathbf{f_a}))$ *in time* $d^{O(r^4)}$ *with probability* $1 - \frac{d^{O(r^4)}}{|S|}$.

Combining theorem 25 and propositions 26 and 28 we have

Lemma 30. *Let* $B(\mathbf{X}) \in (\mathbb{F}[\mathbf{X}])^{s\times t}$ *be a matrix, where* $\mathbf{X} = \{X_1, \ldots, X_n\}$. *Assume the degrees of the entries in* $B(\mathbf{X})$ *to be bounded by* d. *Let* $r \leq \min(s, t)$ *be an integer. Assume that* $\mathtt{V}(\mathrm{Minors}(B(\mathbf{X}), r))$ *is zero-dimensional. Then we can find the set of points* $\mathbf{x} \in \mathtt{V}(\mathrm{Minors}(B(\mathbf{X}), r))$ *in randomized time* $(st \cdot \binom{dr+n+1}{n})^{O(1)} \cdot (dr)^{O(n)}$. *In particular, if* n *is a constant then the running time is bounded by* $(stdr)^{O(1)}$.

A more complicated application of this idea, as suggested by Lemma 24 and used in Lemma 51, is as follows. Let $C(\mathbf{A}, \mathbf{V}) \subseteq (\mathbb{F}[\mathbf{A}][\mathbf{V}])^{s\times t}$ be a polynomial matrix in two sets of variables $\mathbf{A}$ and $\mathbf{V}$. Suppose $s \leq t$, $|\mathbf{A}| = a$ and $|\mathbf{V}| = b+1$, and the degree bound c and d, respectively. We are interested

in computing $\mathbf{a} \in \mathbb{F}^a$ such that $\dim(\text{Minors}(C(\mathbf{a}, \mathbf{V}), s)) = 0$ – namely, all minors of size s in $C(\mathbf{a}, \mathbf{V})$ are zero polynomials in $\mathbb{F}[\mathbf{V}]$. This can be done using Proposition 28 with a bit more work. After getting a $\mathbb{F}$-basis $\mathbf{g}$ of $\text{Minors}(C(\mathbf{A}, \mathbf{V}), s)$, for every $p(\mathbf{A}, \mathbf{V}) \in \mathbf{g}$, view p as in $\mathbb{F}[\mathbf{A}][\mathbf{V}]$, and collect all the coefficients which are polynomials in $\mathbb{F}[\mathbf{A}]$ to form a set R. Now the claim is that $\mathbf{a} \in \mathtt{V}(R)$ if and only if $\mathbf{a}$ sets all polynomials in $\text{Minors}(C(\mathbf{A}, \mathbf{V}), s)$ to zero in $\mathbb{F}[\mathbf{A}][\mathbf{V}]$, which follows by the same proof as in Proposition 28. In this case the number of polynomials we get is bounded by

$$\binom{a+b+s(c+d)}{a+b} \cdot \binom{b+sd}{b} = (a+b+s(c+d))^{O(a+b)} \cdot (b+sd)^{O(b)}.$$

The important observation is that only the number of variables stands on the exponent. We present a proposition summarizing the above discussion.

Proposition 31. *Given a polynomial matrix $C(\mathbf{A}, \mathbf{V}) \in (\mathbb{F}[\mathbf{A}, \mathbf{V}])^{s \times t}$ ($s \leq t$), suppose $|\mathbf{A}| = a$, $|\mathbf{V}| = b + 1$, and the degree bound c and d for $\mathbf{A}$ and $\mathbf{V}$, respectively. Then there exists a randomized algorithm, that computes $g_1, \ldots, g_u \in \mathbb{F}[\mathbf{A}]$, with the following property. If the algorithm succeeds, the g_i's satisfy: $\mathbf{a}$ sets every $s \times s$ minor to a zero polynomial in $\mathbb{F}[\mathbf{V}]$, if and only if $\mathbf{a} \in \mathtt{V}(g_1, \ldots, g_u)$.*

Furthermore, the degree of g_i's is $O(\binom{a+sc-1}{a-1})$, the running time (thus u) is bounded by $O(\binom{a+b+s(c+d)}{a+b} \cdot \binom{b+sd}{b})$, and the success probability is $1 - \frac{(c+d)su}{|S|}$.

4.3 An algorithm extracting the top dimensional component of an ideal

Suppose we are given some polynomials $f_1, \ldots, f_m$ generating $\mathfrak{I} \subseteq \mathbb{F}[X_0, \ldots, X_r]$. Note that in this section we will think of We will be interested in computing the top dimensional component $\text{top}(\mathfrak{I})$. While there are several algorithms to extract the top dimensional component (cf. [DGP98]), we here present a conceptually simple algorithm that suffices in our setting. The purpose is to illustrate the techniques with minimal background. For background material we will mostly refer to [GP07].

We need some ring-theoretic preliminaries to support this algorithm. First are some standard definitions for a Noetherian ring R. For a multiplicative closed subset $M \subseteq R$, $0 \notin M$, $M^{-1}R$ is the *ring of fractions*, where elements are of the form rm^{-1}, $r \in R$ and $m \in M$ with certain (natural) equivalence relation. In the polynomial ring case, we will be mostly interested when $R = \mathbb{F}[\mathbf{X}]$ and $M = \mathbb{F}[\mathbf{U}] \setminus \{0\}$ where $\mathbf{U} \subseteq \mathbf{X}$. We will use $\mathbb{F}[\mathbf{X}]_{\mathbf{U}}$ to denote this case. Also, for an ideal $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$, if $\mathfrak{I} \cap \mathbb{F}[\mathbf{U}] = \{0\}$, then variables in $\mathbf{U}$ are said to be *independent* w.r.t. $\mathfrak{I}$, and $\mathbf{U}$ is called an *independent set* (w.r.t. $\mathfrak{I}$). An independent set $\mathbf{U}$ is *maximal* if $|\mathbf{U}| = \dim(\mathfrak{I})$.

The following fact tells us that for a polynomial ideal of dimension d, there exists d variables that are independent. In fact, this can be taken as another definition of the dimension.

Proposition 32 (Theorem 3.5.1 (6) in [GP07])**.** *For $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$ of dimension d, there exists $\mathbf{U} \subseteq \mathbf{X}$ of size d, such that $\mathbf{U}$ is independent w.r.t. $\mathfrak{I}$.*

Let $S \subseteq R$ be commutative rings, then R is a *finite extension* of S if R is finitely generated as an S-module. That is, there exist $b_1, \ldots, b_m \in R$ such that every $b \in R$ can be expressed as $b = \sum_i a_i b_i$ for $a_i \in S$. For polynomial ideals $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$ of dimension D, *Noether normalization lemma* states that there exist $Y_1, \ldots, Y_D \in \mathfrak{I}$, such that $\mathfrak{I} \cap \mathbb{F}[Y_1, \ldots, Y_D] = \{0\}$, and $\mathfrak{I}$ is a finite extension of $\mathbb{F}[Y_1, \ldots, Y_D]$. We call $\mathfrak{I}$ is in *Noether position* w.r.t. Y_i's. We state the following form of Noether normalization lemma of use to us.

Lemma 33 (Noether normalization lemma. Cf. Theorem 3.4.1 in [GP07])**.** *If $\mathbb{F}$ is infinite, for an ideal $\mathfrak{I} \in \mathbb{F}[\mathbf{X}]$, there exist $M = (m_{ij}) \in \text{GL}(r+1, \mathbb{F})$ and $v = (v_i) \in \mathbb{F}^{r+1}$, such that under the isomorphism $\phi : (X_1, \ldots, X_n) \to (Y_1, \ldots, Y_n)$ where $Y_j = \sum_{i \in [n]} m_{ji} X_i + v_j$, $\phi(\mathfrak{I})$ is in Noether position w.r.t. $Y_1, \ldots, Y_D$.*

Furthermore, such (M, v) can be chosen from a Zariski-dense set in m_{ij}'s and v_i's.

The most important property of Noether normalization for us, is that it puts $Y_1, \ldots, Y_D$ as independent variables for all primary components in $\mathrm{top}(\mathfrak{I})$. In the following, when we say that $X_1, \ldots, X_D$ are put in Noether position w.r.t. an ideal $\mathfrak{I}$, it is understood that we apply some linear transformation to send X_i's to Y_i's such that the situation in the lemma holds.

For an ideal $I \subseteq R$, the *extension* of I to $M^{-1}R$, denoted as I^e, is the ideal in $M^{-1}R$ generated by I in $M^{-1}R$. For an ideal J in $M^{-1}R$, the *contraction* of J w.r.t. R, denoted as J^c is $J \cap R$. Extension and contraction serve as the basic operations for extracting the primary components of an ideal.

Lemma 34 (Proposition 4.3.1 in [GP07])**.** *For $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$ with $\dim(\mathfrak{I}) = D$, suppose the primary decomposition of $\mathfrak{I}$ is $\bigcap_i \mathfrak{J}_i$. If $\mathbf{U}$ is of size D and independent w.r.t. $\mathfrak{I}$, then $\mathfrak{I}^{ec}$ between $\mathbb{F}[\mathbf{X}]$ and $\mathbb{F}[\mathbf{X}]_{\mathbf{U}}$ is the intersection of the primary components $\mathfrak{J}_i$'s satisfying: (1) $\dim(\mathfrak{J}_i) = D$; (2) $\mathbf{U}$ is an independent set for $\mathfrak{J}_i$.*

Based on the above preparation, we present the simple procedure computing $\mathrm{top}(\mathfrak{I})$ for $\mathfrak{I} \subseteq \mathbb{F}[\mathbf{X}]$ of dimension D, codimension $C = r + 1 - D$.

- $\mathfrak{J} := \langle 1 \rangle$.
- Put $X_1, \ldots, X_D$ in Noether position w.r.t. $\mathfrak{I}$.
- $U := \{X_1, \ldots, X_D\}$.
- $\mathfrak{J} := \mathfrak{I}^{ec}$ where extension and contraction are between $\mathbb{F}[\mathbf{X}]$ and $\mathbb{F}[\mathbf{X}]_{\mathbf{U}}$.
- Return $\mathfrak{J}$.

Theorem 35. *Suppose $f_1, \ldots, f_m$ generate an ideal $\mathfrak{I} \subseteq \mathbb{F}[X_0, \ldots, X_r]$ of codimension c, with $\deg(f_i) \leq d$, and $m = O(r)$. Then there is a randomized algorithm computing $g_1, \ldots, g_u$ generating $\mathrm{top}(\mathfrak{I})$ in time $D^{2^{O(r)}}$, with success probability $1 - d^{O(r^2)}/|S|$.*

Proof. The algorithm is just as above; its correctness follows from Lemma 34 and Proposition 32, and the property of Noether normalization. What is left is to exhibit an implementation achieving the time bound as in the statement. It is seen that the above procedure requires the following operations: Noether normalization and extension and contraction. As far as we understand, G. Hermann's classical method [Her26] (cf. [Her98]) is more suitable for worst-case analysis, while Gröbner basis is of course used more widely in practice. We indicate the possibility of using Hermann's method to implement the algorithm without referring to Gröbner basis.

To perform Noether normalization is easy: randomly choose $M \in \mathrm{GL}(r+1, \mathbb{F})$ and $v \in \mathbb{F}^{r+1}$ (recall Lemma 33). To analyze the probability of failure, a bound $1 - d^{O(r^2)}/|S|$ can be achieved (cf. Remark 6 in [JS02]).

To compute $\mathfrak{I}^{ec}$ between $\mathbb{F}[\mathbf{X}]$ and $\mathbb{F}[\mathbf{X}]_{\mathbf{U}}$, some theoretic preparation is needed. The *saturation* of $\mathfrak{I}$ w.r.t. h, denoted as $\mathfrak{I} : h^{\infty}$, is $\{g \in \mathbb{F}[\mathbf{X}] \mid \exists m \in \mathbb{N}, h^m g \in \mathfrak{I}\}$. For the polynomial ideal $\mathfrak{I}$, there exists $h \in \mathbb{F}[\mathbf{U}]$ such that $(\mathfrak{I} : h) = \mathfrak{I}^{ec}$. So the original problem is reduced to the following two tasks.

- Compute $h \in \mathbb{F}[\mathbf{U}]$, such that $(\mathfrak{I} : h) = \mathfrak{I}^{ec}$;
- Compute a set of generators for $(\mathfrak{I} : h)$.

The above two tasks can be accomplished with Hermann's classical method for ideal membership problem, as exhibited in [Asc04], and explained in [Asc11]. This gives the doubly exponential bound $d^{2^{O(r)}}$. □

Several remarks. The algorithm here is the same as in Alonso et al. [AMR90], and background material can be found in [BWK93] or [GP07]. While we follow the philosophy of using Hermann's method to implement the algorithm, in practice Gröbner basis technique is more suitable for implementation and has been realized in a number of softwares. The operations like normalization, extension and contraction, etc. are readily supported. Currently, there is no good upper bound for Buchberger's algorithm (cf. [Asc11]). On the other hand, there exists a doubly-exponential-time algorithm to obtain Gröbner basis for lexicographic order as in [KL90], but as far as we understand, it ultimately relies on idea similar to Hermann's.

We also note that there is a distinction between the geometric version and the algebraic version of the equidimensional decomposition problem. By geometric version of this problem, we mean for $\mathfrak{I}$ of dimension d with equidimensional decomposition $\bigcap_i \mathfrak{I}_i$, compute $\mathfrak{J}_0, \ldots, \mathfrak{J}_d$ such that $\sqrt{\mathfrak{J}_i} = \sqrt{\mathfrak{J}_i}$. In [JS02], Jeromino and Sabia presented an algorithm for this geometric version with time bound $d^{O(r)}$; their algorithm outputs arithmetic circuits as the representation. For the algebraic version, most algorithms assume using the Gröbner basis, as surveyed in [DGP98]. As mentioned, the worst case analysis does not favor Gröbner basis, while in practice it is popular.

5 Low dimensional Formula Reconstruction

In this section, we present the most important component of our algorithm which pertains to doing formula reconstruction when the dimensionality of the ambient projective space is a constant r (we will later set $r = 127$). Here the problem is as follows: given a homogeneous $(r+1)$-variate polynomial $f(\mathbf{Y})$ (here $\mathbf{Y} = (Y_0, Y_1, Y_2, \ldots, Y_r)$) which is the output of a random homogeneous $(\mathbf{Y}, \Delta, S)$-ANF formula ϕ, we want to reconstruct a homogeneous $(\mathbf{Y}, \Delta)$-ANF formula for f. For a certain technical reason pertaining to uniqueness, the output of our low dimensional reconstruction algorithm will be the quadratic forms (in an appropriate order) computed at the second last layer of the tree rather than the linear forms at the leaf nodes of ϕ. Thus given the output polynomial f of an unknown homogeneous $(\mathbf{Y}, \Delta)$-ANF formula ϕ our task is to find quadratic forms $q_1, \ldots, q_m$ $(m = 4^{\Delta-1})$ such that

$$f(\mathbf{Y}) = F_{\Delta-1}(q_1, \ldots, q_m).$$

Let the polynomials computed by the four grandchildren of the output node of ϕ be f_1, f_2, f_3, f_4 so that

$$f = f_1 \cdot f_2 + f_3 \cdot f_4,$$

Our aim is to compute the f_i's and then the list of quadratic forms for f is then simply the concatenation of the four lists of quadratic forms corresponding to each f_i. How do we find the f_i's? We will follow the outline given in section 2. Our presentation here is organized as follows. In section 5.1, we give the formal staement of the algorithm. It will be immediate from the description of the algorithm that its running time is at most $|\phi|^{2^{O(r)}}$ but the algorithm will have a small chance of failing. In section 5.2, we show that if the f_i's and their grandchildren satisfy certain algebraic conditions [24] then our algorithm correctly computes the f_i's (upto appropriate scalar multiples and permutation). This allows us to carry out formula reconstruction for f recursively. In section 5.3, we show that the algebraic conditions of section 5.2 are satisfied with high probability when f is the output of a random homogeneous $(\mathbf{Y}, \Delta, S)$-ANF formula. Finally, in section 5.4 we will put all this together to deduce that in the low dimensional situation, random formulas can be reconstructed efficiently.

[24] We give names to these algebraic conditions in order to convey some intuition for them.

5.1 The Low Dimensional Formula Reconstruction Algorithm

The algorithm is given in the accompanying box. From the description of the algorithm it is fairly straightfoward that the running time is at most $|\phi|^{2^{O(r)}}$. In order to explain what is going on we mention what each step of this algorithm is intended to do/supposed to have computed.

(1) The first step, LDR-1, is just the base case for the recursion.

(2) At the end of step LDR-2, $g_1, g_2, \ldots, g_{r+1}$ are supposed to be a generating set for the ideal $\mathfrak{I}(\mathbf{f}) := \langle f_1, f_2, f_3, f_4 \rangle$.

(3) At the end of step LDR-3, the vector space V is supposed to consist of polynomials of the form

$$(\alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 f_4).$$

and a basis of V is supposed to consist of four $\mathbb{F}$-linearly independent polynomials of the above form.

(4) The h_i's computed in step LDR-4 are supposed to be scalar multiples of the f_i's upto ordering. i.e. there exists a permutation $\pi \in S_4$ and nonzero scalars $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{F}$ such that

$$h_i = \alpha_i \cdot f_{\pi(i)}.$$

(5) The $\widetilde{h_i}$'s computed in step LDR-5 are again scalar multiples of the f_i's but additionally the two sets

$$\{(\widetilde{h_1} \cdot \widetilde{h_2}), (\widetilde{h_3} \cdot \widetilde{h_4})\} \text{ and } \{(f_1 \cdot f_2), (f_3 \cdot f_4)\}.$$

are supposed to be equal.

In essence therefore the $\widetilde{h_i}$'s equal the f_i's upto valid scalar multiples and a valid reordering. Finally note that if a polynomial f is computed by a homogeneous $(\mathbf{Y}, \Delta)$-ANF formula then any scalar multiple $\alpha \cdot f$ is also computed by a $(\mathbf{Y}, \Delta)$-ANF formula. Hence assuming each of the steps LDR-2 to LDR-5 does indeed do what it is supposed to and that the recursive call in step LDR-6 succeeds, we obtain $(\mathbf{Y}, \Delta - 1)$-ANF formulas for the $\widetilde{h_i}$'s and therefore a $(\mathbf{Y}, \Delta)$-ANF formula for f as required.

5.2 Algebraic Nondegeneracy conditions for success of the LDR algorithm.

We begin our analysis of the low dimensional formula reconstruction algorithm given above by defining a certain algebraic nondegeneracy condition which we call *formulaic independence* that is a sufficient condition for the success of steps LDR-2, LDR-3 and LDR-5 in the algorithm above. Let $f = f_1 \cdot f_2 + f_3 \cdot f_4$ and $\mathbf{f} := (f_1, f_2, f_3, f_4)$.

Definition 36. The algebraic set $\mathtt{V}_{\mathbf{J}}(\mathbf{g})$. *Let $\mathbf{g} = (g_1(\mathbf{Y}), \ldots, g_k(\mathbf{Y})) \in \mathbb{F}[\mathbf{Y}]$ be a k-tuple of homogeneous polynomials. The algebraic set $\mathtt{V}_{\mathbf{J}}(g_1, \ldots, g_k)$ ($\mathtt{V}_{\mathbf{J}}(\mathbf{g})$ in short) is defined to be the algebraic set which is the set of common zeroes of polynomials in* $\mathrm{Minors}(J((g_1, \ldots, g_k), \mathbf{Y}), k)$. *In other words, $\mathtt{V}_{\mathbf{J}}(\mathbf{g})$ consists of points $\mathbf{y} \in \mathbb{P}^r$ for which the rank of the Jacobian matrix $J(\mathbf{g}, \mathbf{y})$ is less than k.*

Definition 37. *Situation as above. We will say that $\mathbf{f} = (f_1, f_2, f_3, f_4)$ are* formulaically independent *if*

$$\dim(\mathtt{V}(\mathbf{f})) = r - 4 \quad \textit{and} \quad \dim(\mathrm{Sing}(f) \cap \mathtt{V}_{\mathbf{J}}(\mathbf{f})) < (r - 4).$$

We will say that a homogeneous ANF formula ϕ satisfies formulaic independence *at a node v if*

- *if the node v is a $+$ node and,*

Algorithm : Low Dimensional Reconstruction LDR$(f(\mathbf{Y}), \Delta)$
Input: An $(r+1)$-variate homogeneous polynomial $f \in \mathbb{F}[\mathbf{Y}]$ of degree $d = 2^{\Delta}$ given as a list of coefficients.
Output: Either a tuple of $m = 4^{\Delta-1}$ quadratic forms $(q_1, \ldots, q_m)$ each of rank 4 such that $f = F_{\Delta-1}(q_1, q_2, \ldots, q_m)$ or 'Fail'.

LDR-1: If $\Delta = 1$ then return f itself.

LDR-2: Let $\mathrm{Sing}(f)$ be the ideal generated by the first order derivatives of f - i.e. the ideal

$$\left\langle \frac{\partial f}{\partial Y_0}, \frac{\partial f}{\partial Y_1}, \ldots, \frac{\partial f}{\partial Y_r} \right\rangle.$$

Use Proposition 29 to determine the dimension of $\mathrm{Sing}(f)$. If codimension of $\mathrm{Sing}(f))$ is not 4 output 'Fail'. Else compute a set of generators $g_1, g_2, \ldots, g_u$ for the top-dimensional component (of codimension 4) of $\mathrm{Sing}(f)$ using the algorithm of theorem 35.

LDR-3: Compute a basis $\{\widetilde{g}_1, \ldots \widetilde{g}_t\}$ for the vector space $V \subset \mathbb{F}[\mathbf{Y}]$ consisting of all the homogeneous components of degree $d/2$ of each g_i above. If $t = \dim(V) \neq 4$ output 'Fail'.

LDR-4: By solving an appropriate system of polynomial equations in 4 unknowns, compute another basis $\{h_1, h_2, h_3, h_4\}$ of V such that the singularities of each h_i has a component of codimension 4.

LDR-5: By going over all permutations $\pi : [4] \mapsto [4]$, find one such that f is a $\mathbb{F}$-linear combination of $h_{\pi(1)} \cdot h_{\pi(2)}$ and $h_{\pi(3)} \cdot h_{\pi(4)}$. Compute α, β such that

$$f = \alpha \cdot h_{\pi(1)} \cdot h_{\pi(2)} + \beta \cdot h_{\pi(3)} \cdot h_{\pi(4)}$$

Let

$$\widetilde{h_1} = \alpha \cdot h_{\pi(1)}, \quad \widetilde{h_2} = h_{\pi(2)}, \quad \widetilde{h_3} = \beta \cdot h_{\pi(3)}, \quad \widetilde{h_4} = h_{\pi(4)}$$

LDR-6: For each $i \in [4]$, make a recursive call to LowDimReconstruct$(\widetilde{h_i}, \Delta - 1)$ and obtain

$$Q_i := (q_{i1}, q_{i2}, \ldots, q_{i4^{\Delta-2}}) \quad \text{such that } \widetilde{h_i} = F_{\Delta-2}(q_{i1}, q_{i2}, \ldots, q_{i4^{\Delta-2}}).$$

$$\text{Output} \quad Q := Q_1 \circ Q_2 \circ Q_3 \circ Q_4,$$

where '$\circ$' denotes list concatenation.

Algorithm 1: Low Dimensional Formula Reconstruction (LDR).

- *the four polynomials computed at the grandchildren of v are formulaically independent.*

When f_1, f_2, f_3, f_4 are linear forms, they are formulaically independent if and only if they are $\mathbb{F}$-linearly independent. For higher degree forms, formulaic independence is a somewhat more stringent condition. In particular we have

Proposition 38. Formulaic Independence implies Algebraic Independence. *Let f_1, f_2, f_3, f_4 be $(r+1)$-variate homogeneous polynomials which are formulaically independent. Then they are algebraically independent as well. In particular, $(f_1 \cdot f_2 + f_3 \cdot f_4)$ is a nonzero polynomial.*

Proof. By contradiction. Suppose if possible that the f_i's are algebraically dependent. Then by the Jacobian criterion (13), we have $\text{rank}(J(\mathbf{f})) = 3$ so that each (4×4) minor of $J(\mathbf{f})$ is identically zero. This means that

$$\text{Sing}(f_1 \cdot f_2 + f_3 \cdot f_4) \cap \mathtt{V}(\text{Minors}(J(\mathbf{f}))) = \text{Sing}(f_1 \cdot f_2 + f_3 \cdot f_4).$$

Thus

$$\begin{aligned} \dim(\text{Sing}(f_1 \cdot f_2 + f_3 \cdot f_4) \cap \mathtt{V}(\text{Minors}(J(\mathbf{f})))) &= \dim(\text{Sing}(f_1 \cdot f_2 + f_3 \cdot f_4)) \\ &\geq \dim(\mathtt{V}(\mathbf{f})) \quad (\text{since } \mathtt{V}(\mathbf{f}) \subseteq \text{Sing}(f_1 \cdot f_2 + f_3 \cdot f_4)) \\ &\geq (r-4), \end{aligned}$$

thereby contradicting the formulaic independence of the f_i's. □

In the next subsection we will see that when f_1, f_2, f_3, f_4 are ANF formulas chosen independenty at random then they are formulaically independent with high probability. Next, we give a decomposition of $\text{Sing}(f)$.

Lemma 39.

$$\text{Sing}(f) = \mathtt{V}(\mathbf{f}) \cup (\text{Sing}(f) \cap \mathtt{V}_{\mathbf{J}}(\mathbf{f})). \tag{6}$$

Proof. The decomposition of $\text{Sing}(f)$ as given in equation (6) follows from

$$\text{Sing}(f) \setminus \mathtt{V}(\mathbf{f}) \subseteq \mathtt{V}_{\mathbf{J}}(\mathbf{f}).$$

To prove this containment, it is enough to show that for any $\mathbf{y} \in \text{Sing}(f) \setminus \mathtt{V}(\mathbf{f})$, $J(\mathbf{f}, \mathbf{y})$ is of rank < 4. Note that we have

$$\frac{\partial f}{\partial X_i} = f_1 \cdot \frac{\partial f_2}{\partial X_i} + f_2 \cdot \frac{\partial f_1}{\partial X_i} + f_3 \cdot \frac{\partial f_4}{\partial X_i} + f_4 \cdot \frac{\partial f_3}{\partial X_i}.$$

So if $\frac{\partial f}{\partial X_i}(\mathbf{y}) = 0$ for all $i \in \{0, 1, \ldots, r\}$ then we must have that

$$\begin{bmatrix} \frac{\partial f_1}{\partial X_1}(\mathbf{y}) & \frac{\partial f_2}{\partial X_1}(\mathbf{y}) & \cdots & \frac{\partial f_4}{\partial X_1}(\mathbf{y}) \\ \frac{\partial f_1}{\partial X_2}(\mathbf{y}) & \frac{\partial f_2}{\partial X_2}(\mathbf{y}) & \cdots & \frac{\partial f_4}{\partial X_2}(\mathbf{y}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial X_n}(\mathbf{y}) & \frac{\partial f_2}{\partial X_n}(\mathbf{y}) & \cdots & \frac{\partial f_4}{\partial X_n}(\mathbf{y}) \end{bmatrix} \cdot \begin{bmatrix} f_2(\mathbf{y}) \\ f_1(\mathbf{y}) \\ f_4(\mathbf{y}) \\ f_3(\mathbf{y}) \end{bmatrix} = \mathbf{0}$$

written compactly as

$$J(\mathbf{f}, \mathbf{y})_{(r+1)\times 4} \cdot \mathbf{f}(\mathbf{y})_{4\times 1} = 0.$$

So if $\mathbf{y}$ is not on the algebraic set $\mathtt{V}(\mathbf{f})$ defined by the equations

$$f_1(\mathbf{Y}) = f_2(\mathbf{Y}) = f_3(\mathbf{Y}) = f_4(\mathbf{Y}) = 0$$

then $\mathbf{f}(\mathbf{y})$ is a nonzero vector whence it follows that $J(\mathbf{f}, \mathbf{y})$ has rank at most 3. □

Proposition 40. Formulaic Independence implies Absolute Irreducibility. *Let f_1, f_2, f_3, f_4 be $(r+1)$-variate homogeneous polynomials of degree $d/2$ each. Suppose that they are formulaically independent. Then $f = f_1 \cdot f_2 + f_3 \cdot f_4$ must be absolutely irreducible.*

Proof. By contradiction. Suppose if possible that

$$f(\mathbf{X}) = g(\mathbf{X}) \cdot h(\mathbf{X}), \tag{7}$$

where $g, h \in \mathbb{F}[\mathbf{X}]$ are polynomials of degree d_g and $d_h = (d - d_g)$ respectively.

Claim 41. *g and h must both be homogeneous polynomials.*

Proof of Claim 41: Let the homogeneous decomposition of g and h be

$$g = \sum_{i \in [0..d_g]} g^{[i]} \quad \text{and } h = \sum_{i \in [0..d_h]} h^{[i]}$$

respectively. Then from equation (7) we have

$$f^{[i]} = \sum_{j \in [0..i]} g^{[j]} \cdot h^{[i-j]}$$

where $f^{[i]}$ is the homogeneous component of degree i of f. Now consider $i_g \in [0..d_g]$ and $i_h \in [0..d_h]$ defined as

$$i_g = \min\{i \in [0..d_g] : g^{[i]} \neq 0\}, \quad i_h = \min\{i \in [0..d_h] : g^{[i]} \neq 0\}.$$

Then we have

$$\begin{aligned} f^{[i_g+i_h]} &= g^{[i_g]} \cdot h^{[i_h]} \\ &\neq 0 \end{aligned}$$

Since f is homogeneous of degree d we have get that $i_g + i_h = d$. This can happen if and only if $i_g = d_g$ and $i_h = d_h$ which means that both g and h must be homogeneous. □

Now the fact that $f = g \cdot h$ also means that $\mathrm{Sing}(f) \supseteq \mathtt{V}(g, h)$. By claim 41 above $\mathtt{V}(g, h)$ is a projective variety so that by fact 19 we have $\dim(\mathtt{V}(g, h)) \geq (r-2)$. This means that $\dim(\mathrm{Sing}(f)) \geq (r-2)$ as well. By the above lemma this means that either $\dim(\mathtt{V}(\mathbf{f})) \geq (r-2)$ or $\dim(\mathrm{Sing}(f) \cap \mathtt{V}_{\mathbf{J}}(\mathbf{f})) \geq (r-2)$, both of which contradict the formulaic independence of $\mathbf{f}$. Thus f must be absolutely irreducible. □

With the decomposition of lemma 39 in hand, we are ready to prove that LDR-2, LDR-3 and LDR-5 work correctly assuming formulaic independence of $\mathbf{f}$.

Lemma 42. Correctness of step LDR-2. *If (f_1, f_2, f_3, f_4) are formulaically independent polynomials of degree $\frac{d}{2}$, then step LDR-2 computes a set of polynomials $g_0, g_1, \ldots, g_u$ generating $\mathfrak{J}(\mathbf{f})$, in time $d^{2^{O(r)}}$, with success probability $1 - \frac{d^{O(r^4)}}{|S|}$.*

Proof. Let $\mathfrak{J}$ be the highest equidimensional component of $\mathrm{Sing}(f)$, i.e. $\mathfrak{J} = \mathrm{top}(\mathrm{Sing}(f))$. By the formulaic independence of $\mathbf{f}$ and using the decomposition of $\mathrm{Sing}(f)$ as in lemma 39, we have

(1) $\dim(\mathrm{Sing}(f)) = r - 4$;

(2) for an irreducible component C in $\mathrm{Sing}(f)$, C is of dimension $< (r-4)$ if and only if $C \subseteq \mathrm{Sing}(f) \cap \mathtt{V}_{\mathbf{J}}(\mathbf{f})$.

Given these conditions, we have $\mathfrak{J} = \langle \mathbf{f} \rangle$. By theorem 35, a set of defining polynomials $\mathbf{g} = (g_0, g_1, \ldots, g_u)$ of $\mathfrak{J}$ can be computed in time $d^{2^{O(r)}}$, .i.e. in time $d^{2^{O(r)}}$ we obtain $\mathbf{g} \in (\mathbb{F}[\mathbf{Y}])^u$, where $u = d^{2^{O(r)}}$, such that

$$\langle \mathbf{g} \rangle = \mathfrak{J} = \mathfrak{I}(\mathbf{f}).$$

□

Lemma 43. Correctness of step LDR-3. *Let (f_1, f_2, f_3, f_4) be homogeneous formulaically independent polynomials of degree $d/2$ each. Let*

$$U(\mathbf{f}) := \{\alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 f_4 \ : \ \alpha_i \in \mathbb{F} \ \ \forall i \in [4]\} \subseteq \mathbb{F}[\mathbf{Y}].$$

Then the dimension of $U(\mathbf{f})$ as an $\mathbb{F}$-vector space equals 4 and moreover given a basis $g_0, g_1, \ldots, g_u$ of $\mathfrak{I}(\mathbf{f})$, the algorithm of step LDR-3 computes a basis of $U(\mathbf{f})$ in time $d^{2^{O(r)}}$.

Proof. By proposition 38, the assumption that the f_i's are formulaically independent implies that they are algebraically independent which in turn means that the f_i's are $\mathbb{F}$-linearly independent as well. Thus

$$\dim(U(\mathbf{f})) = 4$$

as an $\mathbb{F}$-vector space. We will use the following notation from section 3. For a polynomial $h \in \mathbb{F}[\mathbf{Y}]$, $h^{[i]}$ shall denote the homogeneous component of degree i of h. Note that in step LDR-3 we compute a basis for $V \subseteq \mathbb{F}[\mathbf{Y}]$ viewed as an $\mathbb{F}$-vector space generated by the homogeneous components of degree $\frac{d}{2}$ of the g_j's, i.e.

$$V := \{\alpha_0 g_0^{[d/2]} + \alpha_1 g_1^{[d/2]} + \ldots + \alpha_{r+1} g_{r+1}^{[d/2]} \ : \ \alpha_i \in \mathbb{F} \ \ \forall i \in [0..(r+1)]\} \subseteq \mathbb{F}[\mathbf{Y}]$$

It suffices then to prove the following claim.

Claim 44.

$$V = U(\mathbf{f}).$$

Proof of Claim 44: Consider an arbitrary polynomial $g(\mathbf{Y}) \in \mathfrak{I}(\mathbf{f})$

$$g(\mathbf{Y}) = \sum_{i \in [4]} h_i(\mathbf{Y}) \cdot f_i(\mathbf{Y}).$$

So the homogeneous component of degree $(d/2)$ of g is

$$\begin{aligned} g^{[d/2]}(\mathbf{Y}) &= \sum_{i \in [4]} (h_i(\mathbf{Y}) \cdot f_i(\mathbf{Y}))^{[d/2]} \\ &= \sum_{i \in [4]} \sum_{0 \le k \le d/2} h_i^{[d/2-k]} \cdot f_i^{[k]} \\ &= \sum_{i \in [4]} h_i^{[0]} \cdot f_i^{[d/2]} \ \ (\text{since } f_i^{[k]} = 0 \ \ \forall k \neq d/2) \\ &= \sum_{i \in [4]} \alpha_i \cdot f_i \ \ \text{where } \alpha_i = h_i^{[0]} \in \mathbb{F}. \end{aligned}$$

Thus $g^{[d/2]} \in U(\mathbf{f})$ for every $g \in \mathfrak{I}(\mathbf{f})$ and hence we have

$$V \subseteq U(\mathbf{f}) \tag{8}$$

Moreover from the same reasoning as above it follows that

$$g^{[k]} = 0 \text{ for every } k < d/2 \text{ and for every } g \in \mathfrak{I}(\mathbf{f}). \tag{9}$$

Since the g_j's form a basis for $\mathfrak{I}(\mathbf{f})$ we have that for each $i \in [4]$, the polynomial f_i can be expressed as

$$f_i = \sum_{0 \le j \le (r+1)} h_j(\mathbf{Y}) \cdot g_j(\mathbf{Y}),$$

for some $h_0, h_1, \ldots, h_{r+1} \in \mathbb{F}[\mathbf{Y}]$. Since f_i is homogeneous of degree $d/2$ it follows that

$$\begin{aligned} f_i &= f_i^{[d/2]} \\ &= \sum_{0 \le j \le (r+1)} (h_j \cdot g_j)^{[d/2]} \\ &= \sum_{0 \le j \le (r+1)} \sum_{0 \le k \le (d/2)} (h_j^{[d/2-k]} \cdot g_j^{[k]}) \\ &= \sum_{0 \le j \le (r+1)} h_j^{[0]} \cdot g_j^{[d/2]} \quad \text{(using (9))} \\ &= \sum_{0 \le j \le (r+1)} \alpha_j \cdot g_j^{[d/2]} \quad \text{where } \alpha_j = h_j^{[0]} \in \mathbb{F}. \end{aligned}$$

Hence each $f_i \in V$ whence

$$U(\mathbf{f}) \subseteq V. \tag{10}$$

Combining equations (8) and (10), the claim follows. □

This completes the proof of lemma 43 as well. □

We next show that formulaic independence also suffices to ensure that step LDR-5 works correctly (assuming that step LDR-4 did).

Lemma 45. Correctness of step LDR-5. *Let $\mathbf{f} = (f_1, f_2, f_3, f_4)$ be homogeneous formulaically independent polynomials of degree $d/2$ each. Assuming that the 4-tuple $\mathbf{h} = (h_1, h_2, h_3, h_4)$ computed in step LDR-4 is $\mathrm{PS}(4, \mathbb{F})$-equivalent to $\mathbf{f}$, i.e. there exists a permutation $\sigma : [4] \mapsto [4]$ and nonzero scalars $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{F}$ such that*

$$f_i(\mathbf{Y}) = \alpha_i h_{\sigma(i)}(\mathbf{Y}) \quad \text{for each } i \in [4]. \tag{11}$$

Then

1. *There do exist scalars $\alpha, \beta \in \mathbb{F}$ and a permutation $\pi : [4] \mapsto [4]$ such that*

   $$f = \alpha h_{\pi(1)} \cdot h_{\pi(2)} + \beta h_{\pi(3)} \cdot h_{\pi(4)}.$$

   Moreover for any fixed π there exists at most one pair $(\alpha, \beta) \in \mathbb{F} \times \mathbb{F}$ satisfying the above equation.

2. *Let $\widetilde{\mathbf{h}} := (\widetilde{h_1}, \widetilde{h_2}, \widetilde{h_3}, \widetilde{h_4})$, where $\widetilde{h_i}$'s are as defined in step LDR-5. Then $\widetilde{\mathbf{h}}$ is $\mathrm{TS}(4, \mathbb{F})$-equivalent to $\mathbf{f}$.*

Proof. We start with the following claim.

Claim 46. *The polynomials $\{f_i f_j \ : \ 1 \le i \le j \le 4\}$ are $\mathbb{F}$-linearly independent. Similarly the polynomials $\{h_i h_j \ : \ 1 \le i \le j \le 4\}$ are $\mathbb{F}$-linearly independent as well.*

Proof of Claim 46: By contradiction. Suppose not. Then for $1 \leq i \leq j \leq 4$ there exist $\alpha_{ij} \in \mathbb{F}$ not all zero such that

$$\sum_{1 \leq i \leq j \leq 4} \alpha_{ij} \cdot f_i \cdot f_j = 0.$$

Consider the 4-variate polynomial $A(\mathbf{Z})$ defined as

$$A(Z_1, Z_2, Z_3, Z_4) := \sum_{1 \leq i \leq j \leq 4} \alpha_{ij} \cdot Z_i \cdot Z_j.$$

Since not all the α_{ij}'s are zero it follows that $A(\mathbf{Z})$ is a nonzero polynomial for which

$$A(f_1, f_2, f_3, f_4) \in \mathbb{F}[\mathbf{Y}]$$

is identically zero. Hence the f_i's are algebraically dependent. But formulaic independence of the f_i's implies that they are algebraically also independent (Proposition 38), a contradiction. The 'similarly' part of the claim follows from the fact that $\mathbf{h}$ is $\mathrm{PS}(4, \mathbb{F})$-equivalent to $\mathbf{f}$. □

Now by assumption, we have

$$\begin{aligned} f &= f_1 \cdot f_2 + f_3 \cdot f_4 \\ &= (\alpha_1 h_{\sigma(1)}) \cdot (\alpha_2 h_{\sigma(2)}) + (\alpha_3 h_{\sigma(3)}) \cdot (\alpha_4 h_{\sigma(4)}) \quad (\text{using} \quad (11)) \\ &= (\alpha_1 \alpha_2) h_{\sigma(1)} \cdot h_{\sigma(2)} + (\alpha_3 \alpha_4) h_{\sigma(3)} \cdot h_{\sigma(4)} \end{aligned}$$

Thus taking $\pi := \sigma, \alpha := \alpha_1 \alpha_2, \beta := \beta_1 \beta_2$ we get the existential part of statement 1 of the lemma. Note that if we had

$$f = \alpha h_{\pi(1)} \cdot h_{\pi(2)} + \beta h_{\pi(3)} \cdot h_{\pi(4)} = \alpha' h_{\pi(1)} \cdot h_{\pi(2)} + \beta' h_{\pi(3)} \cdot h_{\pi(4)},$$

then by the $\mathbb{F}$-linear independence of $h_{\pi(1)} \cdot h_{\pi(2)}$ and $h_{\pi(3)} \cdot h_{\pi(4)}$ we would obtain $\alpha = \alpha'$ and $\beta = \beta'$, proving the 'moreover' part as well. Finally from the statement of LDR-5, it is clear that for some $\beta_1, \beta_2, \beta_3, \beta_4 \in \mathbb{F}^*$ we have

$$\widetilde{h_1} = \beta_1 f_{\pi(1)}, \ \widetilde{h_2} = \beta_2 f_{\pi(2)}, \ \widetilde{h_3} = \beta_3 f_{\pi(3)}, \quad \widetilde{h_4} = \beta_4 f_{\pi(4)}$$

and that

$$f = f_1 \cdot f_2 + f_3 \cdot f_4 = \widetilde{h_1} \cdot \widetilde{h_2} + \widetilde{h_3} \cdot \widetilde{h_4}.$$

Thus we have

$$f_1 \cdot f_2 + f_3 \cdot f_4 = (\beta_1 \cdot \beta_2) \cdot (f_{\pi(1)} \cdot f_{\pi(2)}) + (\beta_3 \cdot \beta_4) \cdot (f_{\pi(3)} \cdot f_{\pi(4)}).$$

Hence from claim 46 we have that

$$(\beta_1 \cdot \beta_2) = (\beta_3 \cdot \beta_4) = 1 \ \text{ and } \{\{1,2\},\{3,4\}\} = \{\{\pi(1), \pi(2)\}, \{\pi(3), \pi(4)\}\}.$$

Using the definition of $\mathrm{TS}(4, \mathbb{F})$, it follows that $\widetilde{\mathbf{h}}$ is $\mathrm{TS}(4, \mathbb{F})$-equivalent to $\mathbf{f}$. This proves part 2 of the lemma as well. □

We remark here that in a similar manner as above, it can be shown that formulaic independence of $\mathbf{f} = (f_1, f_2, f_3, f_4)$ suffices to determine $\mathbf{f}$ upto $\mathrm{OG}(4, \mathbb{F})$-equivalence from f. Also it is impossible in general to do any better. Specifically for any group $G \leq \mathrm{OG}(4, \mathbb{F}) \leq \mathrm{GL}(4, \mathbb{F})$, it is impossible in general (i.e. without any further assumption on the structure of the f_i's) to determine $\mathbf{f}$ upto G-equivalence from f. In what follows we will exploit the fact that the f_i's are computed by $(\mathbf{Y}, \Delta - 1)$-ANF formulas and show that if the grandchildren of the f_i's satisfy certain nondegeneracy conditions then we can determine $\mathbf{f}$ upto $\mathrm{PS}(4, \mathbb{F})$-equivalence and thereafter upto $\mathrm{TS}(4, \mathbb{F})$-equivalence as well.

Definition 47. The Iterated Jacobian and the algebraic set $\mathtt{V_I}$. *Let $\mathbf{g}_1, \ldots, \mathbf{g}_k \in (\mathbb{F}[\mathbf{Y}])^\ell$ be ℓ-tuples of homogeneous $(r+1)$-variate polynomials. The* iterated Jacobian *of $(\mathbf{g}_1, \ldots, \mathbf{g}_k)$, denoted $I(\mathbf{g}_1, \ldots, \mathbf{g}_k)$, is defined to be the following matrix: $I(\mathbf{g}_1, \ldots, \mathbf{g}_k) \in \mathbb{F}[\mathbf{Y}]^{\binom{r+1}{k} \times \ell^k}$ has its rows indexed by k-sized subsets of indices of variables $\{j_1, \ldots, j_k\} \in \binom{[0..r]}{k}$ and its columns indexed by tuples $(i_1, \ldots, i_k) \in [\ell]^k$. The $(\{j_1, \ldots, j_k\}, (i_1, \ldots, i_k))$-th entry of $I(\mathbf{g}_1, \ldots, \mathbf{g}_k, \mathbf{Y})$ is the polynomial*

$$\begin{vmatrix} \frac{\partial g_{1i_1}}{\partial Y_{j_1}} & \frac{\partial g_{2i_2}}{\partial Y_{j_1}} & \cdots & \frac{\partial g_{ki_k}}{\partial Y_{j_1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{1i_1}}{\partial Y_{j_r}} & \frac{\partial g_{2i_2}}{\partial Y_{j_k}} & \cdots & \frac{\partial g_{ki_k}}{\partial Y_{j_k}} \end{vmatrix}.$$

The algebraic set $\mathtt{V_I}(\mathbf{g}_1, \ldots, \mathbf{g}_k)$ is defined to be the common zeroes of the polynomials in $\mathrm{Minors}(I(\mathbf{g}_1, \ldots, \mathbf{g}_k), \ell^k)$. *In other words $\mathtt{V_I}$ is the set of points $\mathbf{y} \in \mathbb{P}^r$ for which the matrix $I(\mathbf{g}_1, \ldots, \mathbf{g}_k, \mathbf{y})$ has rank less than ℓ^k.*

The Situation. In what follows we will look at the following situation:

$$\begin{array}{ll} \text{For } i \in [4] \text{ let } f_i = f_{i1} \cdot f_{i2} + f_{i3} \cdot f_{i4} & \text{and } \forall i, j \in [4] \deg(f_{ij}) = (d/4) \text{ and } f_{ij} \text{ homogeneous} \\ \text{For } i \in [4] \text{ let } \mathbf{f}_i = (f_{i1}, f_{i2}, f_{i3}, f_{i4}) & \text{and } f = f_1 \cdot f_2 + f_3 \cdot f_4 \end{array} \tag{12}$$

For $S = \{i_1, \ldots, i_k\} \subseteq [4]$, let

$$\mathtt{W}_S := \mathtt{V_J}(f_{i_1}, \ldots, f_{i_k}) \cap \mathtt{V_I}(\mathbf{f}_{i_1}, \ldots, \mathbf{f}_{i_k}).$$

Definition 48. *Situation as above. We will say that $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ are* pairwise singularly independent *if*

1. *Dimension of intersection of singularities of any pair of f_i's is at most $(r-6)$, i.e.*

$$\dim(\mathrm{Sing}(f_i) \cap \mathrm{Sing}(f_j)) \leq (r-6) \quad \textit{for all } 1 \leq i < j \leq 4 \quad \textit{and}$$

2. *For all $S \subset [4], |S| \geq 2$:*

$$\dim(\mathtt{W}_S) \leq (r-6).$$

We will say that a homogeneous ANF formula ϕ satisfies pairwise singular independence *at a node v if*

- *if the node v is a + node and,*
- *$(\mathbf{f}_{v_1}, \mathbf{f}_{v_2}, \mathbf{f}_{v_3}, \mathbf{f}_{v_4})$ is pairwise singularly independent where v_1, v_2, v_3, v_4 are nodes which are the grandchildren of v and $\mathbf{f}_{v_i}$ is the 4-tuple of polynomials computed at the grandchildren of the node v_i.*

We will later see that when the formulas computing the f_{ij}'s are independent random ANF formulas then $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ is pairwise singularly independent with high probability. The remainder of this section is devoted to showing that if the above nondegeneracy condition holds true then step LDR-4 of the algorithm works correctly and recovers $\mathbf{f}$ upto $\mathrm{PS}(4, \mathbb{F})$-equivalence.

Lemma 49. *Let $S = \{i_1, \ldots, i_k\} \subseteq [4]$ with $|S| = k \geq 2$. Let $g = \sum_{j \in S} \alpha_j f_j$ where each $\alpha_j \in \mathbb{F}$ is nonzero. If $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ is pairwise singularly independent then*

$$\dim(\mathrm{Sing}(g)) \leq (r-6).$$

Proof. By symmetry, we can assume without assume loss of generality that $S = \{1, \ldots, k\}$.

Claim 50. *We have*

$$\text{Sing}(g) \subseteq \bigcup_{T \subseteq [k]} \left(\mathtt{W}_T \cap \bigcap_{i \in [k] \setminus t} \text{Sing}(f_i) \right) \bigcup \left(\bigcap_{i \in [k]} \text{Sing}(f_i) \right) \tag{13}$$

Proof of Claim 50: Consider an arbitrary $\mathbf{y} \in \text{Sing}(g)$. In order to prove the claim it suffices to show that $\mathbf{y}$ belongs to one of the component varieties on the rhs of equation (13). Towards this end, let $T \subseteq [k]$ consist of all those indices i such that $\mathbf{y} \notin \text{Sing}(f_i)$. If T is empty then $\mathbf{y} \in \bigcap_{i \in [k]} \text{Sing}(f_i)$ and hence we are done. So assume T is nonempty. Assume without loss of generality that $T = \{1, \ldots, t\}$. $t = 1$ is not possible for we have

$$f_1(\mathbf{Y}) = \alpha_1^{-1} \cdot (g(\mathbf{Y}) - \sum_{i \in [2..k]} \alpha_i f_i(\mathbf{Y}))$$

and hence any $\mathbf{y} \in \text{Sing}(g) \cap \cap_{i \in [2..k]} \text{Sing}(f_i)$ is in $\text{Sing}(f_1)$ as well. So we must have that $t \geq 2$. It now suffices to show that $\mathbf{y} \in \mathtt{W}_T$. For this we will need to show that

$$\mathbf{y} \in \mathtt{V}_{\mathbf{I}}(\mathbf{f}_1, \ldots, \mathbf{f}_t) = \mathtt{V}(\text{Minors}(M(\mathbf{f}_1, \ldots, \mathbf{f}_t, \mathbf{Y}), 4^t))$$

where the matrix $M(\mathbf{f}_1, \ldots, \mathbf{f}_t, \mathbf{Y}) \in \mathbb{F}^{\binom{r+1}{t} \times 4^t}$ is the iterated Jacobian matrix as defined in definition 47. For this it suffices to exhibit an explicit nonzero vector in the nullspace of $M(\mathbf{f}_1, \ldots, \mathbf{f}_t, \mathbf{y})$. We will exhibit this vector now. As $\mathtt{V}(\mathbf{f}_i) \subseteq \text{Sing}(f_i)$ (recall $\mathtt{V}(\mathbf{f}_i) = \mathtt{V}(f_{i1}, f_{i2}, f_{i3}, f_{i4})$ by definition) we have that $\mathbf{y} \notin \bigcup_{i \in [t]} \mathtt{V}(\mathbf{f}_i)$. In particular,

$$\exists i_1, \ldots, i_t \in [4] \text{ such that } \forall r \in [t] : f_{ri_r}(\mathbf{y}) \neq 0. \tag{14}$$

Let $\mathbf{v} \in \mathbb{F}^{4^t}$ be the vector indexed by tuples $(i_1, \ldots, i_t) \in [4]^t$, whose $(i_1, \ldots, i_t)$-th entry is $\prod_{r \in [t]} f_{ri_r}(\mathbf{y})$. In other words,

$$\mathbf{v} = \begin{pmatrix} f_{11}(\mathbf{y}) \cdot \ldots f_{t1}(\mathbf{y}) \\ f_{11}(\mathbf{y}) \cdot \ldots f_{t2}(\mathbf{y}) \\ \vdots \\ f_{14}(\mathbf{y}) \cdot \ldots f_{t4}(\mathbf{y}) \end{pmatrix}_{4^t \times 1}$$

From (14) it then follows that $\mathbf{v}$ is a non-zero vector. This will essentially be the non-zero vector that we are looking for. As noted above, it suffices to show that $\mathbf{v}$ is in the nullspace of the matrix $M(\mathbf{f}_1, \ldots, \mathbf{f}_t, \mathbf{y})$. To see this let us look at the equations defining the algebraic set $\text{Sing}(g)$. Note that since $g = \sum_{i \in [k]} \alpha_i f_i$, $\text{Sing}(g)$ is defined by the equations

$$\sum_{i \in [k]} \alpha_i \frac{\partial f_i}{\partial Y_0} = \sum_{i \in [k]} \alpha_i \frac{\partial f_i}{\partial Y_1} = \ldots = \sum_{i \in [k]} \alpha_i \frac{\partial f_i}{\partial Y_r} = 0.$$

From the definition of T we have $\mathbf{y} \in \text{Sing}(f_i)$ for all $i > t$ so we get that

$$\frac{\partial f_i}{\partial Y_j}(\mathbf{y}) = 0 \quad \forall i > t, j \in [0..r].$$

And so using the previous equation we get

$$\sum_{i \in [t]} \alpha_i \frac{\partial f_i}{\partial Y_0}(\mathbf{y}) = \sum_{i \in [t]} \alpha_i \frac{\partial f_i}{\partial Y_1} = \ldots = \sum_{i \in [t]} \alpha_i \frac{\partial f_i}{\partial Y_r}$$

Rearranging this into a matrix form we have

$$J(f_1, f_2, \ldots, f_t, \mathbf{y}) \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_t \end{pmatrix} = \mathbf{0}.$$

As α_i's are non-zero, it follows that for all $\{j_1, \ldots, j_t\} \in \binom{[0..r]}{t}$ we have

$$\begin{vmatrix} \frac{\partial f_1}{\partial Y_{j_1}} & \frac{\partial f_2}{\partial Y_{j_1}} & \cdots & \frac{\partial f_t}{\partial Y_{j_1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial Y_{j_t}} & \frac{\partial f_2}{\partial Y_{j_t}} & \cdots & \frac{\partial f_t}{\partial Y_{j_t}} \end{vmatrix} (\mathbf{y}) = 0. \tag{15}$$

and so $\mathbf{y} \in \mathtt{V_J}(f_1, \ldots, f_t)$. Now we have $f_i = f_{i1} \cdot f_{i2} + f_{i3} \cdot f_{i4}$ so that

$$\frac{\partial f_i}{\partial Y_j} = \sum_{r \in [4]} f_{ir} \cdot \frac{\partial (f_{i\pi(r)})}{\partial Y_j} \ ,$$

where $\pi : [4] \mapsto [4]$ is the map $\pi(1) = 2, \pi(2) = 1, \pi(3) = 4, \pi(4) = 3$. Substituting this in equation (15) and arranging the equations corresponding to all $\{j_1, \ldots, j_t\} \in \binom{[0..r]}{t}$ into a matrix form we get that

$$M(\mathbf{f}_1, \ldots, \mathbf{f}_t, \mathbf{y}) \cdot \begin{pmatrix} f_{1\pi(1)}(\mathbf{y}) \cdot \ldots f_{t\pi(1)}(\mathbf{y}) \\ f_{1\pi(1)}(\mathbf{y}) \cdot \ldots f_{t\pi(2)}(\mathbf{y}) \\ \vdots \\ f_{1\pi(4)}(\mathbf{y}) \cdot \ldots f_{t\pi(4)}(\mathbf{y}) \end{pmatrix} = 0.$$

As the above vector is $\mathbf{v}$ with its entries permuted, it means that $M(\mathbf{f}_1, \ldots, \mathbf{f}_t, \mathbf{y})$ is a singular matrix which in turn means that $\mathbf{y}$ is a common zero of all the $4^t \times 4^t$ minors of $M(\mathbf{f}_1, \ldots, \mathbf{f}_t, \mathbf{Y})$. In other words $\mathbf{y} \in \mathtt{V_I}(\mathbf{f}_1, \ldots, \mathbf{f}_t)$ and so $\mathbf{y} \in \mathtt{W}_T$ (recall that by definition $\mathtt{W}_T = \mathtt{V_J}(f_1, \ldots, f_t) \cap \mathtt{V_I}(\mathbf{f}_1, \ldots, \mathbf{f}_t)$). This completes the proof of the claim. □

From the above claim and using the first two parts of fact 19 it follows that

$$\dim(\mathrm{Sing}(g)) \leq \max\{\delta_1, \delta_2\}$$

where

$$\delta_1 = \dim \left(\bigcup_{T \subseteq [k]} \left(\mathtt{W}_T \cap \bigcap_{i \in [k] \setminus t} \mathrm{Sing}(f_i) \right) \right) \quad \text{and } \delta_2 = \dim \left(\bigcap_{i \in [k]} \mathrm{Sing}(f_i) \right).$$

Since $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ is pairwise singularly independent and $k \geq 2$ we get that $\delta_2 \leq (r-6)$. It then suffices to prove that $\delta_1 \leq (r-6)$. Using part (2) of fact 19, it suffices to prove that for every $T \subseteq [k]$

$$\dim \left(\mathtt{W}_T \cap \bigcap_{i \in [k] \setminus T} \mathrm{Sing}(f_i) \right) \leq (r-6).$$

If $|T| \geq 2$ then this follows from the fact that $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ are pairwise singularly independent (via definition 48). If $|T| = 1$ ($T = \{1\}$ say), then $\mathtt{W}_T \subseteq \mathrm{Sing}(f_1)$ and so using fact 19

$$\begin{aligned} \dim \left(\mathtt{W}_T \cap \bigcap_{i \in [k] \setminus T} \mathrm{Sing}(f_i) \right) &\leq \dim(\mathrm{Sing}(f_1) \cap \mathrm{Sing}(f_2) \cap \ldots \cap \mathrm{Sing}(f_k)) \\ &\leq (r-6). \end{aligned}$$

This proves the lemma. □

Lemma 51. Correctness of step LDR-4. *Situation as above. Assume that* $\tilde{\mathbf{g}} = (\tilde{g}_1, \ldots, \tilde{g}_4)$ *computed in step LDR-3 is* $\mathrm{GL}(4, \mathbb{F})$*-equivalent to* $\mathbf{f}$*. If* $(\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4)$ *are pairwise singularly independent then step LDR-4 of the algorithm correctly computes* $\mathbf{h} = (h_1, h_2, h_3, h_4)$ *which is* $\mathrm{PS}(4, \mathbb{F})$*-equivalent to* $\mathbf{f}$ *in time* $d^{O(r)}$*, with probability* $1 - d^{O(r)}/|S|$*.*

Proof. By the statement of step LDR-4 $\mathbf{h}$ is $\mathrm{GL}(4, \mathbb{F})$-equivalent to $\tilde{\mathbf{g}}$ which is in turn $\mathrm{GL}(4, \mathbb{F})$-equivalent to $\mathbf{f}$. Thus $\mathbf{h}$ is $\mathrm{GL}(4, \mathbb{F})$-equivalent to $\mathbf{f}$. In particular each h_i is an $\mathbb{F}$-linear combination of the f_i's. Also by statement of LDR-4 we have that for each $i \in [4]$:

$$\dim(\mathrm{Sing}(h_i)) = (r - 4).$$

Using lemma 49 above, this means that each h_i is in fact a scalar multiple of some f_j which in turn means that $\mathbf{h}$ is in fact $\mathrm{PS}(4, \mathbb{F})$-equivalent to $\mathbf{f}$. Algorithmically, the h_i's are computed as follows. Consider

$$h = (\alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + \alpha_4 f_4). \tag{16}$$

We think of the α_i's as unknowns and solve for them in the following manner. Firstly take all first partial derivatives $\partial_i h$ for $i \in \{0, \ldots, r\}$. Then use Lemma 24 and Proposition 31 to form $g_1, \ldots, g_u \in \mathbb{F}[\alpha_1, \ldots, \alpha_4]$ such that $(\alpha_1, \ldots, \alpha_4) \in \mathtt{V}(g_1, \ldots, g_u)$ if and only if the codimension of $\mathtt{V}(\partial_0 h, \ldots, \partial_r h)$ is ≤ 4. Note that the parameters in Proposition 31 are set as follows: $a = 4$, $b = 4(r+1) - 1$, $c = 1$, d is the degree bound here and $s = d^{O(1)}$ by the construction in Lemma 24; these give $u = d^{O(r)}$. We view g_i's as defining a algebraic set in $\mathbb{P}^3$. By lemma 49 above, this algebraic set has exactly four points in the projective space $\mathbb{P}^3$. Each of these four solutions in the projective space $\mathbb{P}^3$ gives us an h_i. These four points in $\mathbb{P}^3$ are computed in time $d^{O(r)}$ using the algorithm of Theorem 25. These four points give the h_i's via equation (16). This proves the lemma. □

5.3 Random ANF formulas satisfy the nondegeneracy conditions

Overall Strategy. In this section we show that the polynomials computed by random ANF formulas satisfy the algebraic nondegeracy conditions formulated in definitions 37 and 48. Let $S \subseteq \mathbb{F}$, $\mathbf{X} = (X_0, X_1, \ldots, X_r)$ and $\Delta \geq 2$. Let ϕ be a homogeneous $(\mathbf{X}, \Delta)$ ANF formula so that its output polynomial f can be written as $f = F_\Delta(\ell_1, \ldots, \ell_{4^\Delta})$, for some suitable choice of linear forms $\ell_1, \ldots, \ell_{4^\Delta}$ (fact 7). We will now view the coefficients in the ℓ_i's as formal variables and write

$$\ell_i = \sum_{j \in [0..r]} A_{ij} X_j$$

so that each ℓ_i and hence also f becomes a polynomial in $\mathbb{K}[\mathbf{X}]$, where $\mathbf{A} \stackrel{\text{def}}{=} (A_{ij})_{i \in [4^\Delta], j \in [0..r]}$ and $\mathbb{K} \stackrel{\text{def}}{=} \mathbb{F}(\mathbf{A})$ is the corresponding rational function field. Note that each of the nondegeneracy conditions in definitions 37 and 48 require us to show that the dimension of some related algebraic set $\mathtt{V}_\phi \subseteq \mathbb{P}(\mathbb{K})^r$ is less than some number t say. The defining equations for $\mathtt{V}_\phi$ will be derived out of the polynomials computed at some of the internal nodes of ϕ. By lemma 24 there exist polynomials $p_1(\mathbf{A}), \ldots, p_s(\mathbf{A})$ such that the requisite upper bound on $\dim(\mathtt{V}_\phi)$ is satisfied if and only if $p_i(\mathbf{A}) \neq 0$ for some $i \in [s]$, i.e.

$$\dim(\mathtt{V}_\phi) < t \quad \text{if and only if } \exists i \in [s] \quad \text{such that } p_i(\mathbf{A}) \neq 0 \tag{17}$$

If one of the p_i's was indeed nonzero as a polynomial in $\mathbf{A}$ then via an application of the DeMillo-Lipton-Scwarz-Zippel lemma (lemma 8) we would have that

$$\Pr_{\mathbf{a} \in S^{|A|}}[p_i(\mathbf{a}) \neq 0] \geq \left(1 - \frac{\max_{i \in [s]} \deg(p_i(\mathbf{A}))}{|S|}\right)$$

Hence using (17) we deduce that if there exists an ANF formula ψ for which $\dim(\mathtt{V}_\psi) < t$ then we would have $\dim(\mathtt{V}_\phi) < t$ with probability at least $(1 - \frac{\max_{i\in[s]} \deg(p_i(\mathbf{A}))}{|S|})$ over a random choice of a $(\mathbf{X}, \Delta, S)$-ANF formula ϕ. Note that this estimate on $\Pr_\phi[\dim(\mathtt{V}_\phi) < t]$ is independent of the number s of the characterizing polynomials p_i's.

Applying this strategy. We now instantiate the discussion above for the nondegeneracy conditions in definitions 37 and 48. The following two lemmas (whose proofs we defer to section 5.3.1 and section 5.3.2 respectively) prove the existence of $(\mathbf{X}, \Delta)$-ANF formulas for which the respective nondegeneracy condition is satisfied.

Lemma 52. Existence of formulaically independent ANF formulas. *Let $r \geq 31, \Delta \geq 1$ be integers. Let $\mathbf{X} = (X_0, X_1, \ldots, X_r)$. There exist $(\mathbf{X}, \Delta)$-ANF formulas $\phi_1, \phi_2, \phi_3, \phi_4$ computing polynomials f_1, f_2, f_3, f_4 respectively such that $\mathbf{f} = (f_1, f_2, f_3, f_4)$ is formulaically independent.*

Lemma 53. Existence of Pairwise Singularly Independent $(\mathbf{X}, \Delta)$-ANF formulas. *Let $k > 1$, $r \geq 32k - 1$ and $\Delta \geq 1$ be integers. Let $\mathbf{X} = (X_0, X_1, \ldots, X_r)$ Then there exist polynomials $\{f_{ij} : i \in [k], j \in [4]\}$ each computed by a $(\mathbf{X}, \Delta)$-ANF formula ϕ_{ij} such that*

$$\dim(\mathtt{V}_\mathbf{J}(\mathbf{f}) \cap \mathtt{V}_\mathbf{I}(\mathbf{f}_1, \ldots, \mathbf{f}_k)) \leq (r - 6).$$

where $f_i \stackrel{\text{def}}{=} f_{i1} \cdot f_{i2} + f_{i3} \cdot f_{i4}$ $(i \in [k])$, $\mathbf{f}_i \stackrel{\text{def}}{=} (f_{i1} \cdot f_{i2} + f_{i3} \cdot f_{i4})$, and $\mathbf{f} \stackrel{\text{def}}{=} (f_1, f_2, f_3, f_k)$.

Before presenting the proofs of these two lemmas we derive their consequences following the strategy above. In particular we estimate $\max_i(\deg(p_i(\mathbf{A})))$ for the relevant varieties and thereby obtain estimates for the probability that a random $(\mathbf{X}, \Delta, S)$-ANF formula is nondegenerate.

Corollary 54. *Let $r \geq 31, \Delta \geq 1$ be integers and $S \subseteq \mathbb{F}$. Let $\mathbf{X} = (X_0, X_1, \ldots, X_r)$. Then a random homogeneous $(\mathbf{X}, \Delta, S)$-ANF formula ϕ with a + gate at the top satisfies formulaic independence at its output node with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$.*

Proof. Let $\mathbf{A}$ be the set of coefficients of the linear forms occuring in the leaf nodes of ϕ. Let the polynomial at the output of ϕ be f and the grandchildren be $\mathbf{f} = (f_1, f_2, f_3, f_4)$. The coefficients of the f_i's are polynomials of degree $d_1 = (2^{\Delta-1})$ over $\mathbf{A}$. Consequently the relevant minors of the matrix of the matrix B of lemma 24 are polynomials of degree $d_2 = (d_1^{O(1)}) \cdot d_1 = 2^{O(\Delta)}$. At least one of these minors is nonzero (by lemma 52 above) and hence by the DeMillo-Lipton-Schwarz-Zippel lemma (lemma 8) the probability that $\dim(\mathtt{V}(\mathbf{f})) = r - 4$ is at least $1 - \frac{2^{O(\Delta)}}{|S|}$. Now the coefficients of the derivatives of f are polynomials of degree $d_3 = (2^\Delta)$ in $\mathbf{A}$. The 4×4 minors of $J(f_1, f_2, f_3, f_4)$ are polynomials of degree $d_4 = 2 \cdot 2^{\Delta-1} = 2^\Delta$ in $\mathbf{A}$. Consequently the minors of the relevant matrix capturing the dimension of $\mathrm{Sing}(f) \cap \mathtt{V}_J(\mathbf{f})$ has degree bounded by $d_5 = (d_4^{O(1)}) \cdot (d_4) = 2^{O(\Delta)}$. At least one of these minors is nonzero (by lemma 52 above) and hence by the DeMillo-Lipton-Schwarz-Zippel lemma (lemma 8) the probability that $\dim(\mathrm{Sing}(f) \cap \mathtt{V}_J(\mathbf{f})) < r - 4$ is at least $1 - \frac{2^{O(\Delta)}}{|S|}$. By the union bound overall the probability that ϕ satisfies formulaic independence at its output node is at least $(1 - \frac{2^{O(\Delta)}}{|S|}) = (1 - \frac{|\phi|^{O(1)}}{|S|})$.

□

Corollary 55. *Let $r \geq 127, \Delta \geq 1$ be integers and $S \subseteq \mathbb{F}$. Let $\mathbf{X} = (X_0, X_1, \ldots, X_r)$. Then a random homogeneous $(\mathbf{X}, \Delta, S)$-ANF formula ϕ with a + gate at the top satisfies pairwise singular independence at its output node with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$.*

Proof. The proof is similar to the proof of corollary 54. We estimate the degrees of the relevant polynomials and apply the DeMillo-Lipton-Schwarz-Zippel lemma (lemma 8) to deduce that ϕ satisfies pairwise singular independence at its output node with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$. □

Corollary 56. *Let $\Delta \geq 2, r \geq 31$ be inetegrs and $S \subseteq \mathbb{F}$. Let $\mathbf{X} = (X_0, \ldots, X_r)$. Then the output polynomial of a random homogeneous $(\mathbf{X}, \Delta, S)$-ANF formula ϕ with a + gate at the top is absolutely irreducible with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$. As a consequence, for a random homogeneous ANF formula with a $\times$ gate at the top, the polynomials computed by the two children of the output node are absolutely irreducible with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$.*

Proof. By corollary 54, ϕ satisfies formulaic independence at its output node with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$. By Proposition 40 this means that the output polynomial of ϕ is absolutely irreducible with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$. □

5.3.1 Formulaic Independence of random formulas

Proposition 57. *Let $\mathbf{f} = (f_1, \ldots, f_k) \in \mathbb{F}[\mathbf{X}]$ be a k-tuple of homogenous polynomials. If the f_i's are variable-disjoint then*

$$\mathtt{V}_{\mathbf{J}}(\mathbf{f}) = \bigcup_{i \in [k]} \mathrm{Sing}(f_i).$$

Proof. From the variable disjointness of the f_i's we see that the rows of the Jacobian matrix $J(f_1, \ldots, f_k, \mathbf{x})$ are on disjoint supports and therefore $J(f_1, \ldots, f_k, \mathbf{x})$ is singular if and only if one of its row is entirely zero. This happens if and only if

$$\mathbf{x} \in \bigcup_{i \in [k]} \mathrm{Sing}(f_i).$$

□

Proposition 58. *Let $f_1, \ldots, f_k \in \mathbb{F}[\mathbf{X}]$ be homogenous polynomials, $S = \sum_{i \in [k]} f_i$ and $P = \prod_{i \in [k]} f_i$. Then:*

1. *If the f_i's are variable-disjoint then $\mathrm{Sing}(S) = \bigcap_{i \in [k]} \mathrm{Sing}(f_i)$*
2. *$\mathrm{Sing}(P) = \bigcup_{i \in [k]} \mathrm{Sing}(f_i) \cup \bigcup_{\{i,j\} \in \binom{[k]}{2}} \mathtt{V}(f_i, f_j)$.*

Proof. (1) follows in a straightforward fashion by computing $\frac{\partial S}{\partial X_j}$ and observing that this equals $\frac{\partial f_i}{\partial X_j}$, where f_i is the polynomial that depends on X_j. Thus

$$\{\frac{\partial S}{\partial X_j}\} = \bigcup_{i \in [k]} \{\frac{\partial f_i}{\partial X_j} \ : \ f_i \quad \text{depends on } X_j\}$$

and hence

$$\mathrm{Sing}(S) = \bigcap_{i \in [k]} \mathrm{Sing}(f_i).$$

(2) ($\supseteq$) straightforward.
($\subseteq$) Let P be of degree d and $\mathbf{x} = (x_0, \ldots, x_r) \in \mathrm{Sing}(P)$. So we have,

$$\prod_{i \in [k]} f_i(\mathbf{x}) = P(\mathbf{x}) = \frac{1}{d} \sum_{i \in [n]} x_i (\frac{\partial P}{\partial X_i})(\mathbf{x}) = 0$$

Hence, $\exists \ell \in [k]$ s.t. $f_\ell(\mathbf{x}) = 0$. Now,

$$\forall i \in [n]: \quad (\frac{\partial P}{\partial X_i})(\mathbf{x}) \ = \ \sum_{j \in [k]} (\frac{\partial f_j}{\partial X_i})(\mathbf{x}) \prod_{\substack{r \in [k], \\ r \neq j}} f_r(\mathbf{x}) \ = \ (\frac{\partial f_\ell}{\partial X_i})(\mathbf{x}) \prod_{\substack{r \in [k], \\ r \neq \ell}} f_r(\mathbf{x}) \ = \ 0.$$

Now if $\nexists m \in [k]$ s.t. $m \neq \ell$ and $f_m(\mathbf{x}) = 0$ then $\prod_{r\in[k],r\neq\ell} f_r(\mathbf{x}) \neq 0$ and hence

$$\forall i \in [n] : \ (\frac{\partial f_\ell}{\partial X_i})(\mathbf{x}) \ = \ 0$$

which implies $\mathbf{x} \in \text{Sing}(f_\ell)$. □

Combining this with fact 19 we get:

Corollary 59. *If $f_1, \ldots, f_k$ are pairwise variable disjoint then*

$$\text{codim}(\text{Sing}(f_1 + \ldots + f_k)) = \sum_{i\in[k]} \text{codim}(\text{Sing}(f_i))$$

Proof of lemma 52 : Recall that formulaic independence (definition 37) consists of two parts: (1) $\dim(\mathtt{V}(\mathbf{f})) = r - 4$; (2) $\dim(\text{Sing}(f) \cap \mathtt{V}_\mathbf{J}(\mathbf{f})) < \dim(\text{Sing}(f))$. Let $e = 2^{\Delta-1}$. The specific f_i's we construct we use is:

$$f_i := (X^e_{8(i-1)} + X^e_{8(i-1)+1}) \cdot (X^e_{8(i-1)+2} + X^e_{8(i-1)+3}) + (X^e_{8(i-1)+4} + X^e_{8(i-1)+5}) \cdot (X^e_{8(i-1)+6} + X^e_{8(i-1)+7}).$$

Note that f_i's can be computed by $(\mathbf{X}, \Delta)$-ANF formulas. (1) is satisfied, because of the disjointness of the variables. We verify (2) as follows. By proposition 57 we have

$$\mathtt{V}_\mathbf{J}(\mathbf{f}) = \bigcup_{i\in[4]} \text{Sing}(f_i). \tag{18}$$

Then by Proposition 58 we see

$$\text{Sing}(f) = (\text{Sing}(f_1) \cup \text{Sing}(f_2) \cup \mathtt{V}(f_1, f_2)) \bigcap (\text{Sing}(f_3) \cup \text{Sing}(f_4) \cup \mathtt{V}(f_3, f_4)). \tag{19}$$

We can compute $\text{codim}(\text{Sing}(f_i))$ by applying Proposition 58 to decompose the structure of $\text{Sing}(f_i)$ and then applying corollary 59 we get

$$\text{codim}(\text{Sing}(f_i)) = 4 \quad \forall i \in [4].$$

Combining this with the variable disjointness of the f_i's and applying fact 19 we have

$$\begin{aligned} &\text{codim}(\text{Sing}(f_i) \cap \mathtt{V}(f_j, f_k)) = 6 && \forall i \in [4] \quad \forall j < k \in ([4] \setminus \{i\}) \\ &\text{codim}(\text{Sing}(f_i) \cap \text{Sing}(f_j)) = 8 && \forall i \in [4] \quad \forall j \in ([4] \setminus \{i\}). \end{aligned} \tag{20}$$

Now combining (18) and (19) and simplifying the resulting expression we have

$$\begin{aligned} \text{Sing}(f) \cap \mathtt{V}_\mathbf{J}(\mathbf{f}) \ = \ &\left(\bigcup_{i\in[2]} \text{Sing}(f_i) \cap \mathtt{V}(f_3, f_4)\right) \cup \left(\bigcup_{i\in[3..4]} \text{Sing}(f_i) \cap \mathtt{V}(f_1, f_2)\right) \\ &\cup \left(\bigcup_{i\in[2],j\in[3..4]} \text{Sing}(f_i) \cap \text{Sing}(f_j)\right) \end{aligned}$$

Combining this structural decomposition with (20) and applying fact 19 we have

$$\dim(\text{Sing}(f) \cap \mathtt{V}_\mathbf{J}(\mathbf{f})) \leq (r - 6).$$

This proves the lemma. □

5.3.2 Pairwise Singular Independence of random formulas

Proposition 60. *Let $\mathbf{f}_1 = (f_{11}, \ldots, f_{14}), \ldots, \mathbf{f}_k = (f_{k1}, \ldots, f_{k4})$ be 4-tuples of homogeneous polynomials which are pairwise variable disjoint. Then*

$$\mathtt{V_I}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k) = \bigcup_{i\in[k], j\in[4]} \mathrm{Sing}(f_{ij})$$

Proof. First a bit of notation. We will denote by $\mathrm{var}(f_{ij})$ the set of indices of the variables in f_{ij}, i.e.

$$\mathrm{var}(f_{ij}) \stackrel{\text{def}}{=} \{k \in [0..r] \ : \ \frac{\partial f_{ij}}{\partial X_k} \neq 0\}.$$

Now let $M(\mathbf{f}_1, \ldots, \mathbf{f}_k)$ be the iterated Jacobian matrix as in definition 47. We note that for any $\{j_1, \ldots, j_k\} \in \binom{[0..r]}{k}$ and $(i_1, \ldots, i_k) \in [4]^k$, the $(\{j_1, \ldots, j_k\}, (i_1, \ldots, i_k))$-th entry of $M(\mathbf{f}_1, \ldots, \mathbf{f}_k)$ is

$$\begin{vmatrix} \frac{\partial f_{1i_1}}{\partial X_{j_1}} & \frac{\partial f_{2i_2}}{\partial X_{j_1}} & \cdots & \frac{\partial f_{ki_k}}{\partial X_{j_1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{1i_1}}{\partial X_{j_k}} & \frac{\partial f_{2i_2}}{\partial X_{j_k}} & \cdots & \frac{\partial f_{ki_k}}{\partial X_{j_k}} \end{vmatrix} = \begin{cases} \mathrm{sgn}(\sigma) \prod_{r\in[k]} \frac{\partial f_{ri_r}}{\partial X_{j_{\sigma(r)}}} & \text{if } \exists \sigma \in S_k \text{ s.t. } \forall r \in [k] : j_{\sigma(r)} \in \mathrm{var}(f_{ri_r}) \\ 0 & \text{otherwise} \end{cases}$$

As f_{ij}'s are variable disjoint, every row of M has at most one nonzero entry.
Part 1: $\bigcup \mathrm{Sing}(f_{ij}) \subseteq \mathtt{V_I}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k)$. By symmetry it suffices to show that

$$\mathrm{Sing}(f_{11}) \subseteq \mathtt{V_I}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k).$$

Towards this end consider an arbitrary $\mathbf{x} \in \mathrm{Sing}(f_{11})$. The set of non-zero entries in the first column of $M(\mathbf{f}_1, \ldots, \mathbf{f}_k)$ (upto multiplication by -1) is given by

$$\left\{ \prod_{r\in[k]} \frac{\partial f_{r1}}{\partial X_{j_r}}(\mathbf{X}) \ : \ (j_1, \ldots, j_k) \in \mathrm{var}(f_{11}) \times \ldots \times \mathrm{var}(f_{k1}) \right\}$$

As $\forall j_1 \in \mathrm{var}(f_{11}) \ : \ \frac{\partial f_{11}}{\partial X_{j_1}}(\mathbf{x}) = 0$ we have than the entire first column of $M(\mathbf{f})$ vanishes at $\mathbf{x}$ and hence so do its $4^k \times 4^k$ minors. Thus $\mathbf{x} \in \mathtt{V_I}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k)$, as required.
Part 2: $\mathtt{V_I}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k) \subseteq \bigcup \mathrm{Sing}(f_{ij})$. It suffices to show that if

$$\mathbf{x} \in \mathtt{V_I}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_k) \quad \text{and} \quad \mathbf{x} \notin \bigcup_{i\in[k], j\in[4], (i,j)\neq(1,1)} \mathrm{Sing}(f_{ij}) \tag{21}$$

then $\mathbf{x} \in \mathrm{Sing}(f_{11})$. The second part of (21) means that

$$\forall (i,j) \in ([k] \times [4]) \setminus \{(1,1)\} \ \exists \ell_{ij} \in \mathrm{var}(f_{ij}) \ : \ \frac{\partial f_{ij}}{\partial X_{\ell_{ij}}}(\mathbf{x}) \neq 0$$

Fix any such tuple $\ell_{12}, \ldots, \ell_{14}, \ell_{21}, \ldots, \ell_{k4}$. Consider any $\ell_{11} \in \mathrm{var}(f_{11})$. Now if $\mathbf{x} \in \mathtt{V_I}(\mathbf{f}_1, \ldots, \mathbf{f}_k)$ then the minor of $M(\mathbf{f}_1, \ldots, \mathbf{f}_k)$ (upto multiplication with -1) corresponding to the 4^k tuples in $\{(j_1, \ldots, j_k) \ : \ j_i \in \{\ell_{i1}, \ldots, \ell_{i4}\}\}$ is given by

$$\prod_{i\in[k], j\in[4]} \left(\frac{\partial f_{ij}}{\partial X_{\ell_{ij}}}(\mathbf{x}) \right)^{4^{k-1}} = 0$$

and hence by (21) we have $\frac{\partial f_{11}}{\partial X_{\ell_{11}}}(\mathbf{x}) = 0$. As ℓ_{11} was arbitary we have $\mathbf{x} \in \mathrm{Sing}(f_{11})$. This completes the proof of part 2 and hence of the lemma as well. □

Proof of lemma 53 : Let $e = 2^{\Delta-1}$. The f_{ij}'s we construct are as follows.

$$f_{ij}(\mathbf{X}) = \left(X^e_{32(i-1)+8(j-1)} + X^e_{32(i-1)+8(j-1)+1}\right) \cdot \left(X^e_{32(i-1)+8(j-1)+2} + X^e_{32(i-1)+8(j-1)+3}\right) + \left(X^e_{32(i-1)+8(j-1)+4} + X^e_{32(i-1)+8(j-1)+5}\right) \cdot \left(X^e_{32(i-1)+8(j-1)+6} + X^e_{32(i-1)+8(j-1)+7}\right).$$

Clearly, the f_{ij}'s constructed above are computed by $(\mathbf{X}, \Delta)$ formulas. By construction the f_{ij}'s (and hence also the f_i's) are pairwise variable disjoint and hence we can apply propositions 57 and 60 to obtain decompositions of $\mathtt{V_J}(\mathbf{f})$ and $\mathtt{V_I}(\mathbf{f}_1, \ldots, \mathbf{f}_k)$ respectively. We therefore have

$$\mathtt{V_J}(\mathbf{f}) \cap \mathtt{V_I}(\mathbf{f}_1, \ldots, \mathbf{f}_k) \quad = \quad \left(\bigcup_{i\in[k]} \mathrm{Sing}(f_i)\right) \cup \left(\bigcup_{i\in[k], j\in[4]} \mathrm{Sing}(f_{ij})\right)$$

Now applying proposition 58 we get a decomposition of $\mathrm{Sing}(f_i)$ as follows. For each $i \in [k]$

$$\mathrm{Sing}(f_i) \quad = \quad \mathtt{V}(\mathbf{f}_i) \cup \bigcup_{j\in[4]} \mathtt{W}_{ij} \cup \left(\bigcup_{j\in[1..2], k\in[3..4]} \mathrm{Sing}(f_{ij}) \cap \mathrm{Sing}(f_{ik})\right)$$

where

$$\mathtt{W}_{ij} \stackrel{\text{def}}{=} \begin{cases} \mathrm{Sing}(f_{ij}) \cap \mathtt{V}(f_{i3}, f_{i4}) & \text{if } j \in [1..2] \\ \mathrm{Sing}(f_{ij}) \cap \mathtt{V}(f_{i1}, f_{i2}) & \text{if } j \in [3..4] \end{cases}$$

Combining the two decompositions above and simplifying we obtain

$$\mathtt{V_J}(\mathbf{f}) \cap \mathtt{V_I}(\mathbf{f}_1, \ldots, \mathbf{f}_k) \quad = \quad \left(\bigcup_{j\in[4]} \mathtt{W}_{ij}\right) \cup \left(\bigcup_{(i,j)\neq(i',j')\in[k]\times[4]} \mathrm{Sing}(f_{ij}) \cap \mathrm{Sing}(f_{i'j'})\right) \cup \left(\bigcup_{i,i'\in[k], j\in[4]} \mathrm{Sing}(f_{ij}) \cap \mathtt{V}(\mathbf{f}_{i'})\right)$$

Using the variable disjointness of the f_{ij}'s and applying fact 19 we have

$$\begin{aligned} \mathrm{codim}(\mathtt{W}_{ij}) &= 6 \quad \forall i \in [k] \quad \forall j \in [4] \\ \mathrm{codim}(\mathrm{Sing}(f_{ij}) \cap \mathtt{V}(\mathbf{f}_i)) &= 7 \quad \forall i \in [k] \quad \forall j \in [4] \\ \mathrm{codim}(\mathrm{Sing}(f_{ij}) \cap \mathtt{V}(\mathbf{f}_{i'})) &= 8 \quad \forall i \neq i' \in [k] \quad \forall j \in [4] \\ \mathrm{codim}(\mathrm{Sing}(f_{ij}) \cap \mathrm{Sing}(f_{i'j'})) &= 8 \quad \forall (i,j) \neq (i',j') \in [k] \times [4]. \end{aligned} \tag{22}$$

Consequently

$$\mathrm{codim}(\mathtt{V_J}(\mathbf{f}) \cap \mathtt{V_I}(\mathbf{f}_1, \ldots, \mathbf{f}_k)) \leq (r-6),$$

as required. This proves the lemma.

□

5.4 Putting everything together

We now put everything together into a formal statement.

Theorem 61. *Let $\mathbf{Y} = (Y_0, Y_1, \ldots, Y_r)$ and $\Delta \geq 1$ be an integer. Let $m = 4^{\Delta-1}$ and $S \subseteq \mathbb{F}$. Given blackbox access to the output f of a random homogeneous $(\mathbf{Y}, \Delta, S)$-ANF formula ϕ, with probability at least*

$$\left(1 - \frac{|\phi|^{O(1)}}{|S|}\right) \quad (\textit{over the random choice of } \phi),$$

the LDR algorithm of section 5.1 successfully computes a tuple of m quadratic forms $Q = (q_1, q_2, \ldots, q_m) \in (\mathbb{F}[\mathbf{Y}])^m$ such that:

1. *Each $q_i(\mathbf{Y})$ is a quadratic form of rank four.*
2. $$f = F_{\Delta-1}(q_1, \ldots, q_m)$$
3. *If $Q' = (q'_1, \ldots, q'_m)$ is any other m-tuple for which*

   $$f = F_{\Delta-1}(q'_1, \ldots, q'_m)$$

   then Q' is $\mathrm{TS}(m, \mathbb{F})$-equivalent to Q.

Moreover the running time of the LDR algorithm is bounded by $d^{O(r)}$.

Proof. **Correctness (with high probability).** By corollaries 54 and 55 each + node of ϕ is formulaically independent and pairwise singularly independent with probability at least $(1 - \frac{|\phi|^{O(1)}}{|S|})$. By the union bound, *every node of* ϕ is formulaically independent as well as pairwise singularly independent with probability at least $(1 - |\phi| \cdot \frac{|\phi|^{O(1)}}{|S|}) = (1 - \frac{|\phi|^{O(1)}}{|S|})$. Lemmas 42, 43, 51 and 45 then imply that the LDR algorithm will correctly reconstruct the polynomial computed at each node of ϕ (upto an appropriate group of symmetries). Moreover part (2) of lemma 45 shows that when a node v of ϕ satisfies formulaic independence and pairwise singular independence then the polynomials computed at the grandchildren of v are computed upto $\mathrm{TS}(4, \mathbb{F})$ equivalence. Overall, this means that the quadratic forms are computed correctly upto $\mathrm{TS}(m, \mathbb{F})$-equivalence.
Running Time. It follows from lemmas 42, 43, 51 and 45 that steps LDR-1 to LDR-5 can be accomplished in time $d^{2^{O(r)}}$. Thus the recursion for the running time is

$$T(d) = d^{2^{O(r)}} + 4 \cdot T(d/2)$$

which solves out to give an overall running time of

$$d^{2^{O(r)}} = |\phi|^{2^{O(r)}}.$$

□

6 The Formula Reconstruction algorithm

In section 5 we gave an efficient average-case algorithm for reconstruction of random ANF formulas whose running time was polynomial in the size of the hidden formula. This algorithm had however critically exploited two assumptions:

1. The unknown formula ϕ that we are trying to reconstruct is homogeneous.
2. The number of variables is a constant.

In this section we will effectively get rid of these two assumptions. Let n be the number of variables in the general case. We will use the LDR algorithm of section 5 as a subroutine and show that for arbitrary n, making $O(n^2)$ invocations to the LDR algorithm sufices to patch together the answer to the general problem from the answers to the induced subproblems. This will give us overall a running time of $(sn)^{O(1)}$, where s is the size of the unknown formula ϕ that we are trying to reconstruct. Getting rid of the first assumption is relatively straightforward and we do this in section 6.1. Then in section 6.2 we exploit what we call the *Project and Lift technique* due originally to Kaltofen [Kal89] and to Shpilka [Shp07] in order to get rid of the second assumption. To summarize and for the sake of completeness and concreteness, in section 6.3, we state this overall algorithm and together with the formal statement about the running time and probability of success of the algorithm.

6.1 Homogenization

In this section we will see how the general ANF formula reconstruction problem reduces to the homogeneous ANF formula reconstruction problem. This can be done in a rather straightforward and routine manner (as indicated in the response to Q1 in section 2), the only nontrivial part is to simulate access to a homogenized version of the output polynomial f using blackbox access to f itself.

Proposition 62. *Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$. Let $\Delta \geq 1$ be an integer. Let X_0 be a fresh indeterminate and $d \stackrel{\text{def}}{=} 2^{\Delta}$. For a polynomial $f \in \mathbb{F}[\mathbf{X}]$, define*

$$\hat{f} \stackrel{\text{def}}{=} X_0^d \cdot f(\frac{X_1}{X_0}, \frac{X_2}{X_0}, \ldots, \frac{X_n}{X_0}).$$

Then:

1. *$f(\mathbf{X})$ can be computed by an $(\mathbf{X}, \Delta)$-ANF formula if and only if $\hat{f}$ can be computed by a homogeneous $((X_0, X_1, \ldots, X_n), \Delta)$-ANF formula. Moreover, given a $(\mathbf{X}, \Delta)$-ANF formula for f one can easily obtain a homogeneous $((X_0, X_1, \ldots, X_n), \Delta)$-ANF formula for $\hat{f}$ and vice-versa.*
2. *Given blackbox access to f we can efficiently simulate blackbox access to $\hat{f}$ and vice-versa.*

Proof. **(1).** Let $m = 4^{\Delta}$. By fact 7, f is computed by a $(\mathbf{X}, \Delta)$-ANF formula if and only if there exist *affine forms* $\ell_1, \ldots, \ell_m \in \mathbb{F}[\mathbf{X}]$ such that

$$f = F_{\Delta}(\ell_1, \ldots, \ell_m).$$

Meanwhile $\hat{f}$ is computed by a $((X_0, \ldots, X_n), \Delta)$-ANF formula if and only if there exist *linear forms* $\hat{\ell}_1, \ldots, \hat{\ell}_m$ such that

$$f = F_{\Delta}(\hat{\ell}_1, \ldots, \hat{\ell}_m).$$

Indeed this holds under the natural correspondence which maps an affine form $\ell(\mathbf{X})$ to the linear form

$$\hat{\ell} \stackrel{\text{def}}{=} X_0 \cdot \ell(\frac{X_1}{X_0}, \frac{X_2}{X_0}, \ldots, \frac{X_n}{X_0})$$

which has an inverse given by

$$\ell = \hat{\ell}(1, X_1, X_2, \ldots, X_n).$$

This proves the first part.
(2). Let

$$f(\mathbf{X}) = f^{[d]}(\mathbf{X}) + f^{[d-1]}(\mathbf{X}) + \ldots + f^{[0]}(\mathbf{X})$$

where each $f^{[i]}(\mathbf{X})$ is the homogeneous component of degree i of f. We use the following claim from [Kay12] whose proof we reproduce here for the sake of completeness.

Claim 63. *Given blackbox access to $f(\mathbf{X})$ and a point $\mathbf{x} \in \mathbb{F}^n$, we can compute $f^{[i]}(\mathbf{x})$ for each $i \in [0..d]$ in polynomial time.*

Proof of Claim 63: Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and for $\lambda \in \mathbb{F}$ let

$$\lambda \cdot \mathbf{x} = (\lambda \cdot x_1, \lambda \cdot x_2, \ldots, \lambda \cdot x_n).$$

Then we have

$$f(\lambda \cdot \mathbf{x}) = \lambda^d \cdot f^{[d]}(\mathbf{x}) + \lambda^{d-1} \cdot f^{[d-1]}(\mathbf{x}) + \ldots + \lambda^0 \cdot f^{[0]}(\mathbf{x})$$

so that by plugging in $(d+1)$ different values for λ in the above equation, using the oracle for $f(\mathbf{X})$ to obtain each $f(\lambda \cdot \mathbf{x})$ and solving the resulting system of linear equations we obtain $f^{[i]}(\mathbf{x})$ in polynomial time. (The matrix corresponding to this system of linear equations is a Vandermonde matrix so that it always has an inverse.) □

Now note that since f_i is of degree at most d we have

$$\hat{f}(X_0, X_1, \ldots, X_n) = X_0^0 \cdot f^{[d]}(\mathbf{X}) + X_0^1 \cdot f^{[d-1]}(\mathbf{X}) + \ldots + X_0^d \cdot f^{[0]}(\mathbf{X})$$

Using this we can evaluate $\hat{f}$ at any given point $(x_0, x_1, \ldots, x_n)$. This proves the proposition. □

6.2 Reduction to low dimensional reconstruction.

Kaltofen [Kal85] showed how to reduce the problem of factoring an n-variate polynomial to factoring polynomials with a small number of variables. Subsequently Shpilka [Shp07] showed how reconstructing n-variate $\Sigma\Pi\Sigma(2)$ formulas can be reduced to reconstructing $(\log n)^{O(1)}$-variate $\Sigma\Pi\Sigma(2)$ formulas. These reductions follow a common pattern and we call it the *Project and Lift technique*. We now describe this technique as applicable to our situation.

The Project and Lift Technique. Let $\mathbf{X} = (X_0, X_1, \ldots, X_n)$. Let $\Delta \geq 1$ be an integer. By the discussion in the preceding section, we can assume that the unknown formula is homogeneous. Thus our problem is the following: given a homogeneous polynomial f of degree $d = 2^\Delta$, our task is to compute $m = 4^{\Delta-1}$ quadratic forms $(q_1, \ldots, q_m)$ each of rank at most 4 such that

$$f(\mathbf{X}) = F_{\Delta-1}(q_1, q_2, \ldots, q_m), \tag{23}$$

where the q_i's are the polynomials computed at the second-last level of the unknown ANF formula ϕ computing f. We pick a set of $(n+1)$ linearly independent vectors $\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{F}^{n+1}$. Let $r \in [2..n+1]$ be an integer (r will subsequently be chosen to be an appropriate constant). For $r < i < j \leq n$, let $A_{ij} \in \mathbb{F}^{(n+1)\times(n+1)}$ be the matrix whose k-th column ($k \in [0..n]$) is $\delta_{ijk} \cdot \mathbf{a}_k$, where

$$\delta_{ijk} \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } k \in [0..r] \cup \{i, j\} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

Proposition 64. Reconstructing a quadratic form from its projections. *Let $q(\mathbf{X})$ be a quadratic form over the indicated variable set. Then given $\sigma_{A_{ij}}(q)$ for all $\{i, j\} \in \binom{[(r+1)..n]}{2}$, we can efficiently compute the quadratic polynomial $q(\mathbf{X})$ itself.*

Proof. Upto an appropriate change of basis we can assume without loss of generality that $\mathbf{a}_0 = \mathbf{e}_0, \ldots, \mathbf{a}_n = \mathbf{e}_n$ so that

$$\sigma_{A_{ij}}(q) = q(X_0, \ldots, X_r, 0, \ldots, 0, X_i, 0, \ldots, 0, X_j, 0, \ldots, 0)$$

Let the quadratic form q be

$$q(\mathbf{X}) = \sum_{0 \leq i \leq j \leq n} a_{ij} \cdot X_i \cdot X_j.$$

It is now clear that we can read off each coefficient a_{ij} by looking at the coefficient of $X_i \cdot X_j$ in an appropriate projection $\sigma_{A_{ij}}(q)$. □

The means that if for each $\{i,j\} \in \binom{[(r+1)..n]}{2}$ we could somehow compute $\sigma_{A_{ij}}(q_k)$ then we can recover the quadratic polynomial q_k. How do we obtain $\sigma_{A_{ij}}(q_k)$? We look at the polynomial $\sigma_{A_{ij}}(f)$, i.e. we look at the restriction of f to the subspace spanned by the vectors $\mathbf{a}_0, \ldots, \mathbf{a}_r, \mathbf{a}_i, \mathbf{a}_j$. Observe that since $\sigma_{A_{ij}}$ is a homomorphism, from equation (23) we have that

$$\sigma_{A_{ij}}(f) = F_{\Delta-1}(\sigma_{A_{ij}}(q_1), \ldots, \sigma_{A_{ij}}(q_m)).$$

Thus one can potentially obtain $\sigma_{A_{ij}}(q_k)$ (and therefore solve the n-dimensional reconstruction problem) by solving the reconstruction problem for $\sigma_{A_{ij}}(f)$, which is "merely a constant dimensional problem". $\sigma_{A_{ij}}(f)$ being merely a constant-variate polynomial (when r is chosen as a suitable constant), we can use the low dimensional reconstruction algorithm of section 5 to efficiently reconstruct an ANF formula for f and obtain an m-tuple of quadratic forms

$$\mathbf{q}'_{ij} = (q'_{ij1}, \ldots, q'_{ijm})$$

with each q'_{ijk} of rank four such that

$$\sigma_{A_{ij}}(f) = F_{\Delta-1}(\mathbf{q}'_{ij}).$$

Now if it were true that $\mathbf{q}'$ was equal to $\sigma_{A_{ij}}(\mathbf{q})$, i.e. if it were true that

$$(q'_{ij1}, q'_{ij2}, \ldots, q'_{ij2}) = (\sigma_{A_{ij}}(q_1), \sigma_{A_{ij}}(q_2), \ldots, \sigma_{A_{ij}}(q_m)),$$

then we would have obtained $\sigma_{A_{ij}}(q_k)$ for each $k \in [m]$ and would be done by proposition 64. Unfortunately however this is not true. But we can indeed ensure uniqueness upto a certain subgroup of $\mathrm{GL}(m, \mathbb{F})$. In particular, note that by part (3) of theorem 61 we have that

$$\mathbf{q}'_{ij} \quad \text{is } \mathrm{TS}(m, \mathbb{F}) - \text{equivalent to } \sigma_{A_{ij}}(\mathbf{q})$$

Fortunately this uniqueness of solution upto $\mathrm{TS}(m, \mathbb{F})$ suffices to "*patch together*" the $\mathbf{q}'_{ij}$'s into a valid $\mathbf{q}$. In other words we can patch together the formulas for $\sigma_{A_{ij}}(f)$ into a formula for f itself. In the rest of this section how this works, i.e. we show why knowing $\sigma_{A_{ij}}(\mathbf{q})$ upto $\mathrm{TS}(m, \mathbb{F})$-equivalence suffices for our purpose.

Lemma 65. *Let*

$$\mathbf{q} := (q_1, \ldots, q_m) \in (\mathbb{F}[\mathbf{X}])^m$$

be an m-tuple of quadratic polynomials. Assume that for all $1 \leq k < \ell \leq m$: q_k is **not** *a scalar multiple of q_ℓ. Let $T \subseteq \mathbb{F}$. Let $\mathbf{a}_0, \ldots, \mathbf{a}_n \in \mathbb{F}^{n+1}$ be vectors. There is an algorithm, call it PnL, satisfying the following:*

1. *The algorithm takes as input the vectors $\mathbf{a}_0, \ldots, \mathbf{a}_n \in \mathbb{F}^{n+1}$ and m-tuples $\mathbf{q}'_{ij}$ for each $r < i < j \leq n$ with the property that each $\mathbf{q}'_{ij}$ is $\mathrm{TS}(m, \mathbb{F})$-equivalent to $\sigma_{A_{ij}}(\mathbf{q})$. The algorithm either outputs an m-tuple $\mathbf{q}'$ which is $\mathrm{TS}(m, \mathbb{F})$-equivalent to $\mathbf{q}$ or outputs* `Fail` .

2. *Over the random choice of $\mathbf{a}_0, \ldots, \mathbf{a}_n$ chosen uniformly and independently at random from $T^{n+1} \subseteq \mathbb{F}^{n+1}$, the probability that the algorithm outputs* `Fail` *is at most* ().

3. The running time of the algorithm is bounded by $(nm)^{O(1)}$.

Proof. We first fix a reference m-tuple $\mathbf{p} = (p_1, \ldots, p_m) \stackrel{\text{def}}{=} \mathbf{q}'_{(r+1)(r+2)}$. Let A be the $(n+1) \times (n+1)$ matrix whose first $(r+1)$ columns consist of $\mathbf{a}_0, \ldots, \mathbf{a}_r$ and the rest of the columns are zero.

Claim 66. *Over the random choice of* $\mathbf{a}_0, \ldots, \mathbf{a}_n$ *chosen uniformly and independently at random from* $T^{n+1} \subseteq \mathbb{F}^{n+1}$*, with probability at least* $(1 - \frac{4}{|T|})$ *it holds true that* $\sigma_A(q_k)$ *is not a scalar multiple of* $\sigma_A(q_\ell)$ *(*$k \neq \ell$*).*

Proof of Claim 66: Note that two polynomials $q_k(\mathbf{X})$ and $q_\ell(\mathbf{X})$ are scalar multiples of each other if and only if they are $\mathbb{F}$-linearly dependent. The claim then follows by an application of lemma 10. □

Now fix a randomly chosen $\mathbf{a}_0, \ldots, \mathbf{a}_n$ such that the conclusion of the above claim holds true. By making a change of basis if necessary we can assume without loss of generality that $\mathbf{a}_0 = \mathbf{e}_0, \mathbf{a}_1 = \mathbf{e}_1, \ldots, \mathbf{a}_n = \mathbf{e}_n$ so that for any $q(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ we have

$$\sigma_A(q) = q(X_0, \ldots, X_r, 0, \ldots, 0) \quad \text{and } \sigma_{A_{ij}}(q) = q(X_0, \ldots, X_r, 0, \ldots, 0, X_i, 0, \ldots, 0, X_j, 0, \ldots, 0).$$

Now the above claim means in particular that every $\{i, j\} \in \binom{[(r+1)..n]}{2}$ there exists a unique matrix $B_{ij} \in \mathrm{TS}(m, \mathbb{F})$ such that $\sigma_A(\mathbf{p}) = B_{ij} \cdot \sigma_A(\mathbf{q}'_{ij})$ and moreover that each such B_{ij} can be computed in time $m^{O(1)}$. Algorithmically we compute all the B_{ij}'s and replace each $\mathbf{q}'_{ij}$ by $B_{ij} \cdot \mathbf{q}'_{ij}$ so that going forward we can assume that $\sigma_A(\mathbf{p}) = \sigma_A(\mathbf{q}'_{ij})$ for every $\{i, j\} \in \binom{[(r+1)..n]}{2}$. We now apply apply proposition 64 to all the m components of the $\mathbf{q}'_{ij}$'s separately and this gives us the required $\mathbf{q}' \in \mathbb{F}[\mathbf{X}]^m$ which is $\mathrm{TS}(m, \mathbb{F})$-equivalent to the unknown $\mathbf{q} \in \mathbb{F}[\mathbf{X}]^m$. The overall running time is clearly $(nm)^{O(1)}$. □

6.3 The overall algorithm.

With all the necessary tools in hand, we are now state a more formal version of our main theorem and the overall ANF formula reconstruction algorithm.

Theorem 67 (Main)**.** *Let* $\mathbf{X} = (X_1, \ldots, X_n)$*,* $\mathbb{F}$ *be a field of characteristic 0 and* S *be a finite subset of* $\mathbb{F}$*. Let* $\Delta \geq 1$ *be an integer and* $s = 4^\Delta$*. Given blackbox access to the output* f *of a random* $(\mathbf{X}, \Delta, S)$*-ANF formula* ϕ*, with probability at least*

$$\left(1 - \frac{n^2.s^{O(1)}}{|S|}\right) \quad (\textit{over the random choice of } \phi),$$

algorithm 2 successfully computes a tuple of s *affine forms* $L = (\ell_1, \ldots, \ell_s) \in (\mathbb{F}[\mathbf{X}])^s$ *such that*

$$f = F_\Delta(\ell_1, \ldots, \ell_s).$$

Moreover the running time of the algorithm is $\mathsf{poly}(n, s)$*.*

Proof of Theorem 67 : The algorithm is as given in the accompanying box. We now analyze its correctness and running time.

Correctness (with high probability). In step AFR3 each $\sigma_{A_{ij}}(f)$ is a effectively a random $(r+1)$-variate ANF formula so that by theorem 61 the probability that step AFR3 succeeds is at least $\left(1 - \frac{n^2|\phi|^{O(1)}}{|S|}\right)$. Moreover a pair of independent chosen quadratic forms of rank four are $\mathbb{F}$-linearly independent with probability at least $(1 - \frac{4}{|S|})$ so that all the quadratic forms at the second last level

Algorithm: ANF Formula Reconstruction AFR($f(\mathbf{X}), \Delta$)
Input: Blackbox access to an n-variate homogeneous polynomial $f \in \mathbb{F}[\mathbf{X}]$ of degree at most $d = 2^{\Delta}$.
Output: Either a set of 4^{Δ} affine forms $\ell_1, \ldots, \ell_{4^{\Delta}}$ such that $f = F_{\Delta}(\ell_1, \ldots, \ell_{4^{\Delta}})$ or 'Fail'.

AFR1: If $\Delta = 0$ then f is an affine form. Compute f via interpolation and return the affine form.

AFR2: **Homogenization.** Use the algorithm of proposition 62 to homogenize f. So now assume that $\mathbf{X} = (X_0, X_1, \ldots, X_n)$ and f is a homogeneous polynomial of degree exactly $d = 2^{\Delta}$.

AFR3: **Reduction to LDR.** Pick $(n+1)$ vectors $\mathbf{a}_0, \ldots, \mathbf{a}_n$ each of whose coordinates are chosen uniformly at random from a large enough subset $T \subseteq \mathbb{F}$. Let $r = 127$ and $m := 4^{\Delta - 1}$. For $r < i < j \leq n$, let A_{ij} be the $(n+1) \times (n+1)$ matrix whose k-th column ($k \in [0..n]$) is $\delta_{ijk} \cdot \mathbf{a}_k$ where δ_{ijk} is as defined in equation (24). For each A_{ij} invoke the LDR algorithm on $\sigma_{A_{ij}}(f)$ to obtain an m-tuple $\mathbf{q}_{ij} = (q_{ij1}, \ldots, q_{ijm})$ satisfying

- $\mathrm{rank}(q_{ijk}) \leq 4$ for each $i, j \in \binom{[(r+1)..n]}{2}$ and $k \in [m]$ and
- $\sigma_{A_{ij}}(f) = F_{\Delta - 1}(\mathbf{q}_{ij})$.

AFR4: **Patchwork.** Invoke the algorithm of lemma 65 on input $((\mathbf{a}_0, \ldots, \mathbf{a}_n), (\mathbf{q}_{ij})_{r<i<j\leq n})$ and obtain an m-tuple of quadratic forms $\mathbf{q} = (q_1, q_2, \ldots, q_m)$.

AFR5: For each $i \in [m]$, find linear forms $\ell_{i1}, \ell_{i2}, \ell_{i3}, \ell_{i4}$ such that

$$q_i = \ell_{i1} \cdot \ell_{i2} + \ell_{i3} \cdot \ell_{i4}.$$

Output $(\ell_{11}, \ldots, \ell_{14}, \ell_{21}, \ldots, \ell_{m3}, \ell_{m4})$.

Algorithm 2: ANF Formula Reconstruction (AFR)

of ϕ are pairwise $\mathbb{F}$-linearly independent with probability at least $(1 - \frac{4n^2}{|S|})$. By lemma 65 this ensures that step AFR4 with probability at least $(1 - \frac{4n^2}{|S|})$. The remaining steps succeed for any choice of ϕ. Thus over the random choice of a $(\mathbf{X}, \Delta, S)$ formula our algorithm succeeds with probability at least

$$\left(1 - \frac{n^2 \, |\phi|^{O(1)}}{|S|}\right).$$

Running Time. The running time is dominated by step ADR3 and the time taken here is $n^2 \cdot d^{2^{O(r)}}$ which is $n^2 \cdot |\phi|^{O(1)}$ as r is a constant. Overall therefore the running time is bounded by $|\phi|^{O(1)}$. This completes the proof of our main theorem.

□

We conclude with some remarks of a somewhat speculative nature concerning speedup, improvements and generalizations of the above algorithm which may one day make it a practical one.

Remark 68.

1. Currently, the most expensive step of our algorithm is Step LDR-2 of the low dimensional reconstruction algorithm, where we need to apply Theorem 35 to extract the top dimensional component. We note that in practice, the algorithm using Gröbner basis to extract the top dimensional component is readily supported in some softwares like SINGULAR [GP07]. Thus this step should not be considered as expensive from the practical point of view. Theoretically, it may well be that formulaic independence of $\mathbf{f} = (f_1, f_2, f_3, f_4)$ suffices to ensure that $\mathfrak{I}(\mathbf{f})$ is a radical ideal, i.e. $\sqrt{\mathfrak{I}(\mathbf{f})} = \mathfrak{I}(\mathbf{f})$. In this case we would have that $\mathfrak{I}(\mathbf{f}) = \mathfrak{I}(\mathtt{V}(\mathbf{f}))$ (by Hilbert's string Nullstellensatz). In this situation we can use the algorithm of Jeronimo and Sabia [JS02] to extract the top dimensional component of $\mathrm{Sing}(f)$ and using this we would incur a cost in the running time of only $d^{O(r)}$ rather than $d^{2^{O(r)}}$ for the `LDR` algorithm.

2. **Choice of r.** The choice of the parameter r (which is currently set to 127) has a huge impact on the running time of our algorithm. Our present choice of r stems from lemmas 52 and 53. These lemmas may hold true true for much smaller values of r, maybe even for $r = 6$. If so, this can have a significant the practical running time of our algorithm. Combined with the above this means that a running time of $O(d^8) = O(|\phi|^4)$ or maybe even $O(|\phi|^3)$ seems possible.

3. **When the characteristic of the field is small.** The same algorithm as above (possibly with a different choice of the parameter r) remains valid even when the characteristic of the underlying field is small. But there are some significant changes to the analysis in this situation. We indicate these changes. Firstly, the Jacobian criterion of theorem 13 is valid only in one direction when the characteristic of the field is small but this is okay because in our application (in proposition 38) we only this valid direction. Most of the remaining analsysis (except the proofs of lemmas 52 and 53) carry over to the low characteristic situation when we interpret the derivatives $\frac{\partial f}{\partial X_i}$ symbolically. In the proof of lemmas 52 and 53 we use variable-disjoint but isomorphic copies of polynomials of the form

$$f = (X_1^e + X_2^e) \cdot (X_3^e + X_4^e) + (X_5^e + X_6^e) \cdot (X_7^e + X_8^e), \quad \text{where } e = 2^{\Delta - 1}.$$

Over any field $\mathbb{F}$ of characteristic different from two we have $\mathrm{codim}(\mathrm{Sing}(f)) = 4$ so that lemmas 52 and 53 continue to be valid over such fields. When the characteristic of $\mathbb{F}$ is 2 if we instead choose

$$f = (X_1^{e-1} X_2 + X_3^{e-1} X_4) \cdot (X_5^{e-1} X_6 + X_7^{e-1} X_8) + (X_9^{e-1} X_{10} + X_{11}^{e-1} X_{12}) \cdot (X_{13}^{e-1} X_{14} + X_{15}^{e-1} X_{16})$$

($e = 2^{\Delta-1}$) then such an f can be computed by a homogeneous $(\mathbf{X}, \Delta)$ ANF formula and satisfies $\mathrm{codim}(\mathrm{Sing}(f)) = 4$. The rest of the analysis (in particular proofs of propositions 57, 58 and 60) carry over unchanged so that theorems 61 and 1 continue to remain valid even over fields of characteristic two (though with a slightly larger value of r).

4. **Success probability.** Note that when $\ell_1, \ell_2, \ell_3, \ell_4$ are independent n-variate linear forms with coeffcients chosen independently at random from $S \subseteq \mathbb{F}$, the probability that these are $\mathbb{F}$-linearly dependent is at most $p = \frac{4}{|S|}^{n/4}$. (To see this note that the above quantity is the same as the probability that a random $4 \times n$ matrix A has rank 3 when the entries of A are chosen uniformly at random from $S \subseteq \mathbb{F}$. If we now break matrix A into $n/4$ disjoint submatrices of dimension 4×4 each, then the determinants of these submatrices are each nonzero as formal polynomials and these are on disjoint sets of variables. Now if $\mathrm{rank}(A) < 4$ then each of these disjoint submatrices must have determinant zero and by independence this happens with probability at most $\frac{4}{|S|}^{n/4}$.) The nondegeneracy conditions of definitions 37 and 48 are in some sense generalization of $\mathbb{F}$-linear independence to higher degree polynomials. It is quite plausible that an argumnt akin to the one above can show that the success probability of this algorithm is at least

$$\left(1 - \left(\frac{n\,|\phi|}{|S|}\right)^{O(n)}\right).$$

5. **Allowing larger fanin.** It should be possible to generalize the average-case reconstruction algorithm presented here to formulas wherein the internal nodes to have fanin bounded by some constant k (the formula would still have alternating layers of addition and multiplication gates as before).

7 Conclusion

Connection to Lower bounds and an update. In this work we exploit the dimension of the variety defined by the first order partial derivatives of a polynomial to devise the reconstruction algorithm. It was noted in an earlier version of this work that a very weak lower bound can be obtained by looking at the dimesnion of the variety defined by the first-order derivatibes of a polynomial.

Proposition 69. *Let $d \geq 2$ and $n \geq 5$ be integers and let*

$$g = X_1^d + X_2^d + \ldots + X_n^d.$$

Then for any representation of g of the form

$$g = h_{11} \cdot h_{12} + h_{21} \cdot h_{22} + \ldots + h_{s1} \cdot h_{s2},$$

where the h_{ij}'s are homogeneous polynomials, the number of summands s must be at least $\frac{n-1}{2}$.

Proof. It is easily verified that in this case, $\mathrm{Sing}(g)$ is empty as a projective variety and hence dimension of $\mathrm{Sing}(g)$ is 0. On the other hand $\mathrm{Sing}(g)$ contains the variety

$$\mathtt{V} := \mathtt{V}(h_{11}, h_{12}, \ldots, h_{s1}, h_{s2}).$$

Since $\mathtt{V}$ is a projective variety in an ambient space of dimension $n-1$ and is defined by $2s$ polynomials, its dimension is at least $n - 1 - 2s$. Since $\mathrm{Sing}(g)$ contains $\mathtt{V}$ its dimension must be at least as large as that of $\mathtt{V}$, whence we have that $s \geq \frac{n-1}{2}$. □

Pushing these ideas much further, Gupta et al [GKKS12] look at the variety defined by (some of) the higher order derivatives of a polynomial and exploiting a connection between the dimension of a variety and the number of $\mathbb{F}$-linearly independent polynomials in a slice of the ideal corresponding to the variety. In this way a they are able to new lower bounds for depth four circuits.

Hardness of Reconstruction. One of the underlying motivations of this work was to understand the complexity of computing the minimum formula for a polynomial given as a blackbox. We are not aware for a hardness result for this problem and pose it as a conjecture.

Conjecture 70. Finding the smallest formula computing a given polynomial f (for concreteness assume that f itself is given via an arithmetic formula) is NP-hard.

Buchfuhrer and Umans show the hardness of the corresponding problem for Boolean formulas is hard. In particular the following comment from [BU08] seems especially relevant in the context of this conjecture.

> *One reason reductions for these problems are difficult is that one direction of the reduction entails proving a lower bound for the type of circuit under consideration. This shouldnt be an absolute barrier, though, for two reasons. First, we have lower-bound proof techniques for (Boolean) formulas ... nevertheless incorporating these into a reduction seems tricky. Second, a reduction need not entail strong lower bounds and in principle even slightly non-trivial lower bounds could suffice.*

References

[AHM+08] Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael E. Saks. Minimizing Disjunctive Normal Form Formulas and AC^0 Circuits Given a Truth Table. *SIAM J. Comput.*, 38(1):63–84, 2008.

[AMR90] Maria Emilia Alonso, Teo Mora, and Mario Raimondo. Local decomposition algorithms. In *AAECC*, pages 208–221, 1990.

[AMS10] Vikraman Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New results on noncommutative and commutative polynomial identity testing. *Computational Complexity*, 19(4):521–558, 2010.

[Asc04] Matthias Aschenbrenner. Ideal membership in polynomial rings over the integers. *J. Amer. Math. Soc.*, 7:407–411, 2004.

[Asc11] Matthias Aschenbrenner. Algorithms for computing saturations of ideals in finitely generated commutative rings: Appendix to: Automorphisms mapping a point into a subvariety, j. algebraic geom. 20 (2011), 785-794, 2011.

[BBB+00] Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000.

[BC98] Nader H. Bshouty and Richard Cleve. Interpolating arithmetic read-once formulas in parallel. *SIAM J. Comput.*, 27(2):401–413, 1998.

[BCE91] Nader H. Bshouty, Richard Cleve, and Wayne Eberly. Size-depth tradeoffs for algebraic formulae. In *FOCS*, pages 334–341, 1991.

[BOT88] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynominal interpolation (extended abstract). In *STOC*, pages 301–309, 1988.

[BU08] David Buchfuhrer and Christopher Umans. The complexity of boolean formula minimization. In *ICALP (1)*, pages 24–35, 2008.

[BWK93] T. Becker, V. Weispfenning, and H. Kredel. *Gröbner bases: a computational approach to commutative algebra*. Graduate texts in mathematics. Springer-Verlag, 1993.

[CLO05] D.A. Cox, J.B. Little, and D. O'Shea. *Using algebraic geometry*. Graduate texts in mathematics. Springer, 2005.

[CLO07] D.A. Cox, J.B. Little, and D. O'Shea. *Ideals, Varieties and Algorithms*. Undergraduate texts in mathematics. Springer, 2007.

[DGP98] Wolfram Decker, Gert-Martin Greuel, and Gerhard Pfister. Primary decomposition: Algorithms and comparisons, 1998.

[Dub93] Thomas W. Dubé. A combinatorial proof of the effective Nullstellensatz. *J. Symb. Comput.*, 15:277–296, March 1993.

[ER93] Richard Ehrenborg and Gian-Carlo Rota. Apolarity and canonical forms for homogeneous polynomials. *Eur. J. Comb.*, 14(3):157–181, 1993.

[FK06] Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. In *COLT*, pages 350–363, 2006.

[GKKS12] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:98, 2012.

[GKL11] Ankit Gupta, Neeraj Kayal, and Satya Lokam. Efficient reconstruction of random multilinear formulas. In *FOCS*, pages 778–787, 2011.

[GKL12] Ankit Gupta, Neeraj Kayal, and Satya Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In *STOC*, pages 625–642, 2012.

[GP07] G.M. Greuel and G. Pfister. *A Singular Introduction to Commutative Algebra*. Springer, 2007.

[Har92] Joe Harris. *Algebraic Geometry: a first course*. Springer, 1992.

[Hås90] Johan Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11:644–654, December 1990.

[Her26] Grete Hermann. Die frage der endlich vielen schritte in der theorie der polynomideale. *Mathematische Annalen*, 95:736–788, 1926. 10.1007/BF01206635.

[Her98] Grete Hermann. The question of finitely many steps in polynomial ideal theory. *SIGSAM Bull.*, 32(3):8–30, September 1998.

[JS02] Gabriela Jeronimo and Juan Sabia. Effective equidimensional decomposition of affine varieties. *Journal of Pure and Applied Algebra*, 169(2?):229 – 248, 2002.

[Kal85] Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2):469–489, 1985.

[Kal89] Erich Kaltofen. Factorization of polynomials given by straight-line programs. *Randomness and Computation*, 5:375–412, 1989.

[Kay11] Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1409–1421, 2011.

[Kay12] Neeraj Kayal. Affine projections of polynomials. In *STOC*, pages 643–662, 2012.

[KC00] Valentine Kabanets and JinYi Cai. Circuit minimization problem. In *STOC*, pages 73–79, 2000.

[KL90] Teresa Krick and Alessandro Logar. Membership problem, representation problem and the computation of the radical for one-dimensional ideals. In *Proceedings MEGA-90*, 1990.

[KLW10] Adam R. Klivans, Homin K. Lee, and Andrew Wan. Mansour's conjecture is true for random dnf formulas. In *COLT*, pages 368–380, 2010.

[KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *STOC*, pages 216–223, 2001.

[KS03] Adam Klivans and Amir Shpilka. Learning arithmetic circuits via partial derivatives. In *COLT*, pages 463–476, 2003.

[KS09] Zohar Shay Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *IEEE Conference on Computational Complexity*, pages 274–285, 2009.

[KY88] Erich Kaltofen and Lakshman Yagati. Improved sparse multivariate polynomial interpolation algorithms. In *ISSAC*, pages 467–474, 1988.

[Laz81] Daniel Lazard. Resolution des systemes d'equations algebriques. *Theor. Comput. Sci.*, 15:77–110, 1981.

[Laz01] Daniel Lazard. Solving systems of algebraic equations. *SIGSAM Bull.*, 35(3):11–37, September 2001.

[Laz09] Daniel Lazard. Thirty years of polynomial system solving, and now? *J. Symb. Comput.*, 44(3):222–231, 2009.

[Man94] Yishay Mansour. Learning boolean functions via the fourier transform. In *Theoretical Advances in Neural Computation and Learning*, pages 391–424, 1994.

[Sel09] Linda Sellie. Exact learning of random DNF formulas over the uniform distribution. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, pages 45–54, New York, NY, USA, 2009. ACM.

[Shp07] Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. In *STOC*, pages 284–293, 2007.

[SV08] Amir Shpilka and Ilya Volkovich. Read-once polynomial identity testing. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 507–516, New York, NY, USA, 2008. ACM.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010.

[vzG87] J. von zur Gathen. Permanent and determinant. *Linear Algebra and its Applications*, 96:87–100, 1987.

[Yap00] C.K. Yap. *Fundamental problems of algorithmic algebra*. Oxford University Press, 2000.

[Zip90] Richard Zippel. Interpolating polynomials from their values. *J. Symb. Comput.*, 9(3):375–403, 1990.