Electronic Colloquium on Computational Complexity, Report No. 99 (2022)

Equivalence Test for Read-Once Arithmetic Formulas

Nikhil Gupta
Indian Institute of Science
nikhilg@iisc.ac.in

Chandan Saha*
Indian Institute of Science
chandan@iisc.ac.in

Bhargav Thankey†
Indian Institute of Science
thankeyd@iisc.ac.in

Abstract

We study the polynomial equivalence problem for *orbits* of read-once arithmetic formulas (ROFs). Read-once formulas have received considerable attention in both algebraic and Boolean complexity and have served as a testbed for developing effective tools and techniques for analyzing circuits. Two n-variate polynomials $f, g \in \mathbb{F}[\mathbf{x}]$ are *equivalent*, denoted as $f \sim g$, if there is an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f = g(A\mathbf{x})$. The *orbit* of f is the set of all polynomials equivalent to f. We investigate the complexity of the following two natural problems on ROFs:

- *Equivalence test for ROFs:* Given black-box access to f, check if it is in the orbit of an ROF. If yes, output an ROF $\mathtt{C}$ and an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f = \mathtt{C}(A\mathbf{x})$.
- *Polynomial equivalence for orbits of ROFs:* Given black-box access to f and g in the orbits of two *unknown* ROFs, check if $f \sim g$. If yes, output an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f = g(A\mathbf{x})$.

These problems are significant generalizations of two well-studied problems in algebraic complexity, namely reconstruction of ROFs and quadratic form equivalence. In this work, we give the first randomized polynomial-time algorithms (with oracle access to quadratic form equivalence) to solve the two problems. The equivalence test works for *general* ROFs; it also implies an efficient learning algorithm for *random* arithmetic formulas of unbounded depth and fan-in (in the high number of variables setting). The algorithm for the second problem, which invokes the equivalence test, works for mildly restricted ROFs, namely additive-constant-free ROFs.

The equivalence test is based on a novel interplay between the factors and the essential variables of the Hessian determinant of an ROF, the essential variables of the ROF, and certain special structures in the ROF that we call "skewed paths". To our knowledge, the Hessian of a general ROF (or even a depth-4 ROF) has not been analyzed before. Analyzing the Hessian and combining the knowledge gained from it with the skewed paths to recursively discover formulas in the orbits of sub-ROFs of lower depth (without incurring an exponential blow-up due to unbounded depth) constitute the main technical contributions of this work.

*Partially supported by a MATRICS grant of the Science and Engineering Research Board, DST, India.
†Supported by the Prime Minister's Research Fellowship, India.

ISSN 1433-8092

Contents

1 Introduction

The study of isomorphism or equivalence of two mathematical objects, such as graphs, groups, rings, algebras, tensors, polynomials, and formulas, under natural bijective transformations has a rich history. The corresponding computational problems of determining isomorphism between two such entities are not only interesting from a theoretical standpoint but also important for their potential applications in other areas – most notably, in cryptography [Pat96, IQ19, JQSY19].

Our focus, in this work, is on the isomorphism or equivalence of polynomials. Two n-variate polynomials $f(\mathbf{x})$ and $g(\mathbf{x})$ over a field $\mathbb{F}$ are said to be *equivalent* (denoted as $f \sim g$) if they are equal under the action of an invertible linear transformation on the variables, i.e., if $f = g(A\mathbf{x})$ for an $A \in \mathrm{GL}(n, \mathbb{F})$. The *orbit* of f (denoted as $\mathrm{orb}(f)$) consists of all polynomials that are equivalent to f. Equivalent polynomials share many common algebraic and geometric properties. The computational task of checking if two polynomials – given as lists of coefficients – are equivalent is known as the *polynomial equivalence or isomorphism problem* (PE). PE is one of the most important problems in algebraic complexity and is also widely studied in cryptography.

The exact computational complexity of PE remains an enigma despite decades of research. Over finite fields, PE is in $\mathsf{NP} \cap \mathsf{coAM}$ [Thi98, Sax06], and so, it is not NP-complete unless the polynomial hierarchy collapses. But, no subexponential-time algorithm is known for PE over $\mathbb{F}_q$. The best-known complexity of PE over other fields, such as $\mathbb{C}$ and $\mathbb{R}$, is the same as that of checking solvability of a system of polynomial equations, which is a potentially harder problem. Research on PE has therefore focused on understanding the complexity of the problem for restricted (yet interesting) classes of polynomials; see Section 1.4 for a brief account of known results on PE.

1.1 Motivations

1.1.1 Generalizing quadratic form equivalence

One of the very few natural classes of polynomials for which PE is known to be efficiently solvable is the class of quadratic forms or homogeneous polynomials of degree 2 (see Section A.2 for the complexity of quadratic form equivalence over various fields). Are there bigger classes of polynomials for which PE is easy? An obvious way to generalize quadratic form equivalence is to solve PE for higher degree forms. Unfortunately, even cubic form equivalence is at least as hard as graph isomorphism and possibly harder (see Section 1.4 for a discussion on this). But there is another natural way to generalize quadratic form equivalence: Over $\mathbb{C}$, an n-variate quadratic form with *no redundant variables*[1] is in the orbit of $x_1x_2 + x_3x_4 + \ldots + x_{n-1}x_n$, if n is even[2]. The expression $x_1x_2 + x_3x_4 + \ldots + x_{n-1}x_n$ is a read-once arithmetic formula (ROF)[3]. An ROF is an arithmetic formula in which every leaf node is labelled by a distinct variable or a constant (see

[1] i.e., the number of variables cannot be reduced by applying an invertible linear map on the variables.

[2] If n is odd, the quadratic form is in the orbit of $x_1x_2 + \ldots + x_{n-2}x_{n-1} + x_n^2$. The "odd n" case can be reduced to the "even n" case of the quadratic form equivalence problem over $\mathbb{C}$ by simply adding x_{n+1}^2 to both the input forms. The correctness of this reduction follows from Witt's cancellation theorem [Wit37]; see [CMM17] for a self-contained proof.

[3] That $\mathbb{F} = \mathbb{C}$ is *not* very crucial here. Over an arbitrary $\mathbb{F}$ of characteristic $\neq 2$, a quadratic form is in the orbit of some $a_1x_1^2 + \ldots + a_nx_n^2$, where $a_i \in \mathbb{F}$. Observe that $a_1x_1^2 + \ldots + a_nx_n^2$ is an ROF with every x_i replaced by x_i^2. We call a formula a *power-substituted* ROF if it is derived from an ROF by replacing every x_i by $x_i^{e_i}$ for some $e_i \in \mathbb{N}$. Although we work with conventional ROFs in this work, our results and analysis are likely to generalize (after some modifications) to power-substituted ROFs. See Section 6 for an evidence supporting this belief.

Section 2.1.4). The quadratic form equivalence problem (QFE) over $\mathbb{C}$ can thus be viewed as PE for orbits of *quadratic* ROFs. Is PE for orbits of higher degree ROFs easy? More generally, we ask:

Can we solve PE for orbits of general ROFs efficiently?

In other words, is there an efficient algorithm which, when given black-box access to f and g in the orbits of two *unknown* ROFs of unbounded degree and depth, decides if $f \sim g$? Theorem 2 answers this question (almost entirely) positively, thereby implying a vast and rare generalization of efficient quadratic form equivalence.

1.1.2 Learning orbits of well-studied circuit classes

Learning or reconstructing arithmetic circuits is one of the three most fundamental problems in arithmetic circuit complexity alongside proving circuit size lower bounds and polynomial identity testing (or constructing hitting-sets). While learning general circuits or formulas is believed to be a hard problem, significant progress has been made in designing efficient learning algorithms for various interesting special classes of circuits. Sparse polynomials (or depth-2 circuits)[4], ROFs, and read-once algebraic branching programs (ROABPs)[5] are notable instances of such classes (see Section 1.5 in [GKS20] for a brief account of known results on circuit reconstruction).

As two equivalent polynomials are essentially the same function (up to a choice of the coordinate system), it is natural to wonder if the known learning algorithms for the above-mentioned circuit classes can be generalized to work for their orbits[6]. Unfortunately, the techniques used to learn these classes do not extend in a straightforward manner to learning their orbits. So, studying these orbits may lead to strengthening of existing techniques and discovery of new ones in the process. But there is also a deeper reason to investigate orbits of simple-looking circuit classes that originates from a connection between affine projections[7] and orbits.

Affine projections of apparently weak circuit classes can be extremely powerful. For instance, affine projections of sparse polynomials constitute depth-3 circuits – a surprisingly powerful class [GKKS16, Tav15]. Likewise, affine projections of ROFs and ROABPs capture general formulas and ABPs, respectively. It turns out that the affine projections of a polynomial f are contained in the *closure*[8] of $\text{orb}(f)$. In this sense, $\text{orb}(f)$ is a *dense* subset of affine projections of f. Therefore, it is necessary to analyze orbits of the above-mentioned classes to better understand their affine projections. Spurred by these reasons, [MS21], [ST21], and [BG21] have recently given hitting-set constructions for orbits of sparse polynomials, ROFs, and bounded width ROABPs. Can we design learning algorithms for the same orbits? In this work, we answer the question for ROFs. To our knowledge, learning orbits of sparse polynomials and, more generally, ROABPs remain open.

Formally, the learning problem for orbits of ROFs is as follows: Given black-box access to a polynomial f, check if it is in the orbit of an ROF. If yes, then output an invertible A such that $f(A\mathbf{x})$ has an ROF. We call this problem *equivalence test (ET) for ROFs*.

Can we solve equivalence test for ROFs efficiently?

[4] A polynomial is s-sparse if it has at most s monomials with non-zero coefficients.

[5] An ROABP is an expression $\mathbf{1}^T \cdot M_1(x_1) \cdots M_n(x_n) \cdot \mathbf{1}$, where $\mathbf{1}$ is the all-one column vector and each $M_i(x_i)$ is a matrix whose entries are univariate polynomials in x_i. ROABPs generalize sparse polynomials and ROFs significantly.

[6] Orbit of a circuit class $\mathscr{C}$, denoted as $\text{orb}(\mathscr{C})$, is the union of the orbits of circuits in $\mathscr{C}$. Learning $\text{orb}(\mathscr{C})$ amounts to outputting a circuit $C \in \mathscr{C}$ and an invertible A from black-box access to a $f(\mathbf{x}) \in \text{orb}(\mathscr{C})$ such that C computes $f(A\mathbf{x})$.

[7] The set of affine projections of an n-variate polynomial $f(\mathbf{x})$ is the set $\{f(A\mathbf{x}+\mathbf{b}) \ : \ A \in \mathbb{F}^{n\times n} \text{ and } \mathbf{b} \in \mathbb{F}^n\}$.

[8] The closure of $\text{orb}(f)$ is the Zariski closure of the set of coefficient vectors of polynomials in $\text{orb}(f)$.

Theorem 1 answers the question *completely* in the affirmative. ET for ROFs is a substantial generalization of the well-studied problem of learning or reconstructing ROFs [HH91, BHH95a, SV14, Vol16, MV18]. Indeed, the proof of the theorem requires significantly new ideas on top of those used for ROF reconstruction (see Section 1.3).

The above question is closely related to the question posed in Section 1.1.1. A typical algorithm to solve the search version[9] of QFE over $\mathbb{C}$ finds invertible linear transformations that map the two input quadratic forms to the *canonical* ROF $x_1x_2 + x_3x_4 + \ldots + x_{n-1}x_n$. In other words, such an algorithm solves the ET problem for quadratic ROFs. Similarly, our algorithm in Theorem 2 invokes the ET given by Theorem 1 to map the input polynomials to certain canonical ROFs and then solve the PE problem for canonical ROFs.

1.1.3 Learning random or non-degenerate formulas

As mentioned before, a formula is an affine projection of an ROF. Learning formulas in the *worst-case* is a potentially hard problem (see Section 1.2 in [KS19] and Section 1.4 in [GKS20] for discussions on this). However, it may be possible to formulate natural distributions (or non-degeneracy conditions) under which formulas are learnable.[10] A natural distribution on formulas is defined as follows: pick a tree of size s *arbitrarily*, label the internal nodes by $+$ and $\times$ operations to form alternating layers of $+$ and $\times$ gates, and label the leaves by *random* linear forms in n variables. The corresponding learning problem asks to reconstruct a random formula – picked according to this distribution – from black-box access to the formula. This problem was studied in [GKQ14] by *fixing* the underlying tree to be a complete binary tree; the formulas we thus get are called formulas in *alternating normal form* (ANF). [GKQ14] gave an efficient learning algorithm for random ANFs. On the other hand, an ET for ROFs gives a learning algorithm for random formulas, *irrespective of the underlying tree*, provided $n \geq s$. This is because a random formula is in the orbit of an ROF with high probability if $n \geq s$ and $|\mathbb{F}|$ is sufficiently large. Thus, ET for ROFs provides supporting evidence for efficient learnability of random formulas (that are not necessarily ANFs).

1.2 Main results

Our results hold over any field $\mathbb{F}$ of characteristic 0 or of sufficiently large characteristic and size. As for the computation model, we assume that it allows basic field operations in unit time and univariate polynomial factoring in randomized polynomial time. We say an algorithm is efficient if it runs in randomized polynomial time. Also, we will work with a slightly general definition of the orbit of a polynomial that allows translation (see Definition 2.4).

For the ease of stating the theorems, we consider ROFs in *canonical* form (see Definition 2.6). The orbit of every ROF contains a canonical ROF (Observation 2.6). So, by removing redundant variables from the input polynomial (see Observation 2.7 and Claim 2.2), we can assume *without any loss of generality* that the underlying ROF is canonical. $x_1x_2 + x_3x_4 + \ldots + x_{n-1}x_n$ is a canonical ROF. Other natural examples of canonical ROFs are read-once formulas in alternating normal form (ROANF) (see Definition 2.7) and the sum-product polynomial $\mathsf{SP} := \sum_{i\in[s]} \prod_{j\in[d]} x_{i,j}$ [11].

[9] The search version of PE asks to find an $A \in \mathrm{GL}(n, \mathbb{F})$ s.t. $f = g(A\mathbf{x})$, if such an A exists. For the search version of QFE over $\mathbb{C}$, we work with a computation model that allows basic complex arithmetic and square root finding.

[10] Several prior works have given efficient non-degenerate case learning algorithms for various subclasses of formulas for which worst-case learning are likely hard (see [GKS20, KS19, KNS19, GKQ14, GKL11, BGKS21]).

[11] The Boolean analogue of the SP polynomial is a read-once DNF which is also known as the *tribes function*.

Our first result gives an efficient algorithm to solve ET for *general* ROFs. The algorithm is randomized and has oracle access to the search version of QFE. In subsequent discussions, we will mention "QFE" to mean "the search version of QFE". We will also identify an ROF with the polynomial it computes and denote the set of $n \times n$ matrices with entries in $\mathbb{F}$ by $M(n, \mathbb{F})$.

Theorem 1 (ET for ROFs)**.** *Let $n \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n^2$, and $|\mathbb{F}| \geq n^{13}$. There is a $\mathrm{poly}(n)$ time randomized algorithm (with oracle access to QFE over $\mathbb{F}$) that takes input black-box access to an n-variate polynomial $f \in \mathbb{F}[\mathbf{x}]$, which is in the orbit of an unknown canonical ROF $\mathtt{C}$, and outputs (with high probability) an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f(A\mathbf{x}) = \mathtt{C}(PS\mathbf{x} + \mathbf{b})$, where $P \in M(n, \mathbb{F})$ and $S \in M(n, \mathbb{F})$ are permutation and scaling (i.e., diagonal) matrices respectively, and $\mathbf{b} \in \mathbb{F}^n$.*

Remarks.

1. As $\mathtt{C}(PS\mathbf{x} + \mathbf{b})$ is an ROF, we can apply any of the known polynomial-time ROF reconstruction algorithms [HH91, BHH95a, SV14, MV18] to first get an ROF for $\mathtt{C}(PS\mathbf{x} + \mathbf{b})$, and then obtain a formula for f by applying A^{-1} on the variables of the reconstructed ROF. We present a randomized polynomial-time ROF reconstruction algorithm in Appendix F as we need to use some of its properties in the proof of Theorem 2.
2. QFE can be solved efficiently over $\mathbb{C}, \mathbb{R}, \mathbb{F}_q$ and also over $\mathbb{Q}$ with oracle access to integer factoring (see Fact A.3). Hence, ET for ROFs can be solved efficiently over these fields.
3. Although ET has been studied for polynomial families like the determinant and IMM (see Section 1.4), to our knowledge no ET was known for any natural circuit class of unbounded *depth*, *degree* and *fan-in* (or even depth-4 ROFs) before this work.
4. Recently, [MS21] showed that ET for ROANFs and sum-product polynomials can be solved efficiently. As ROANFs are special fan-in 2 ROFs and sum-product polynomials are depth-2 ROFs, the theorem generalizes these two results considerably. Also, our proof approach is entirely different from the ones in [MS21] (see Sections 1.3.1 and 1.3.4).
5. The constraints on $\mathrm{char}(\mathbb{F})$ and $|\mathbb{F}|$ originate primarily (but not solely) from the use of the black-box multivariate polynomial factorization algorithm [KT90] in the equivalence test. We have not made an attempt to optimize these constraints.

The second result gives an efficient algorithm to solve PE for orbits of ROFs that are *additive-constant-free*. An ROF is additive-constant-free if no $\mathbb{F}$-constant appears as a child of a $+$-gate. For e.g., the canonical ROF $x_1x_2 + x_3x_4 + \ldots + x_{n-1}x_n$ is additive-constant-free. ROANFs and sum-product polynomials are also examples of additive-constant-free canonical ROFs. An additive-constant-free ROF is in the orbit of an additive-constant-free canonical ROF (see Observation 2.6).

Theorem 2 (PE for orbits of additive-constant-free ROFs)**.** *Let $n \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n^2$, and $|\mathbb{F}| \geq n^{13}$. There is a $\mathrm{poly}(n)$ time randomized algorithm (with oracle access to QFE over $\mathbb{F}$) that takes input black-box access to two n-variate polynomials $f_1, f_2 \in \mathbb{F}[\mathbf{x}]$, which are in the orbits of two unknown additive-constant-free canonical ROFs, and checks if $f_1 \in \mathrm{orb}(f_2)$. Furthermore, if $f_1 \in \mathrm{orb}(f_2)$, then the algorithm outputs (with high probability) an $A \in \mathrm{GL}(n, \mathbb{F})$ and a $\mathbf{b} \in \mathbb{F}^n$ such that $f_1 = f_2(A\mathbf{x} + \mathbf{b})$.*

Remarks.

1. As mentioned before, the above result is a broad generalization of efficient QFE.
2. We strongly believe that the additive-constant-free restriction is mild and can be dispensed with entirely with some more technical effort. To support this belief, we show in Section E that this indeed possible for depth-4 ROFs.

1.3 Proof ideas

First, an example. The algorithm in Theorem 1 is based on a few crucial properties of the Hessian determinant of an ROF (see Definition 2.9). The effectiveness of the Hessian, in this context, is best demonstrated by an equivalence test for the sum-product polynomial $\mathsf{SP} := \sum_{i\in[s]} \prod_{j\in[d]} x_{i,j}$, which is an ROF of product-depth 1. Assume that $d \geq 3$. The algorithm takes input an $f = \mathsf{SP}(B\mathbf{x})$, where $B \in \mathrm{GL}(sd, \mathbb{F})$ is unknown. It computes the Hessian determinant of f, which is denoted as $\det(H_f)$. By Fact 2.8, $\det(H_f)$ is a non-zero $\mathbb{F}$-multiple of $\det(H_{\mathsf{SP}})(B\mathbf{x})$ – the Hessian determinant of SP evaluated at $B\mathbf{x}$. Now, it can be shown that $\det(H_{\mathsf{SP}})$ factorizes as follows:

$$\det(H_{\mathsf{SP}}) = (-1)^{s(d-1)} \cdot (d-1)^s \cdot \prod_{i\in[s], j\in[d]} x_{i,j}^{d-2}.$$

So, the algorithm factorizes $\det(H_f)$ into irreducible factors and figures out[12] B from the factors. The test can be implemented in the black-box setting by observing that black-box access to the second-order partials of f can be computed efficiently (see Fact A.1) and by invoking a black-box polynomial factorization algorithm (see Fact A.2). The running time is polynomial in s and d.

1.3.1 A basic approach

Can the Hessian determinant be exploited to devise an equivalence test for ROFs of *arbitrary* product-depth and fan-in? A rudimentary approach is outlined in [Kay11]: Let $g = g_1(x_1, \ldots, x_i) + g_2(x_{i+1}, \ldots, x_n)$, where g_1 and g_2 are variable disjoint polynomials. Given black-box access to $f = g(B\mathbf{x})$, where g and $B \in \mathrm{GL}(n, \mathbb{F})$ are unknown, can we find an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f(A\mathbf{x})$ can be expressed as a sum of two variable disjoint polynomials?[13] [Kay11] gave an algorithm that finds such an A *provided the number of essential variables*[14] *of* $\det(H_g)$ *is exactly* n.

The algorithm uses the fact that $\det(H_g) = \det(H_{g_1})(x_1, \ldots, x_i) \cdot \det(H_{g_2})(x_{i+1}, \ldots, x_n)$, and so, $\det(H_f)$ is a non-zero $\mathbb{F}$-multiple of $\det(H_{g_1})(B\mathbf{x}) \cdot \det(H_{g_2})(B\mathbf{x})$. It turns out that an $A \in \mathrm{GL}(n, \mathbb{F})$ can be found efficiently from black-box access to $\det(H_f)$ such that $\det(H_{g_1})(BA\mathbf{x})$ and $\det(H_{g_2})(BA\mathbf{x})$ are variable disjoint; this step involves black-box factorization of $\det(H_f)$ [KT90] and elimination of redundant variables from the irreducible factors of $\det(H_f)$ in a careful way (see Claim 2.4). Now, it can also be shown that if the number of essential variables of $\det(H_g)$ is exactly n, then $g_1(BA\mathbf{x})$ and $g_2(BA\mathbf{x})$ are variable disjoint (see Observation 2.2).

The correctness of the algorithm depends critically on the condition that the number of essential variables of $\det(H_g)$ is *exactly* n. If this condition does not hold, then the algorithm fails completely. The approach can be viewed as a generalization of the algorithm given in the above example for the SP polynomial. Indeed, the number of essential variables of $\det(H_{\mathsf{SP}})$ is $n = sd$.

Can the basic approach be used to learn orbits of ROFs? At a high level, the basic approach is encouraging as a $+$-rooted ROF is a sum of variable disjoint polynomials. Let $\mathsf{C} = T_1 + \ldots + T_s$ be a $+$-rooted canonical ROF, where $T_1, \ldots, T_s$ are the *terms* of C, i.e., the polynomials computed

[12] The algorithm finds an $A = PSB$, where P is a permutation matrix and S is a diagonal matrix, from the factors of $\det(H_f)$. It then interpolates $f(A^{-1}\mathbf{x})$ (using the sparse polynomial interpolation algorithm in [KS01]) to learn P and S (up to the symmetries of the polynomial SP).

[13] This problem was referred to as the *polynomial decomposition* problem in [Kay11]. It should not be confused with the functional decomposition of polynomials which is also known as the polynomial decomposition problem.

[14] See Definition 2.1.

by the second (from the top) layer of gates in $\mathtt{C}$. Given black-box access to $f = \mathtt{C}(B\mathbf{x}) = T_1(B\mathbf{x}) + \ldots + T_s(B\mathbf{x})$, where $\mathtt{C}$ and $B \in \mathrm{GL}(n, \mathbb{F})$ are unknown, we hope to apply the approach in [Kay11] to find an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $T_1(BA\mathbf{x}), \ldots, T_s(BA\mathbf{x})$ are *variable disjoint*. If we succeed in finding A, then we wish to obtain *efficient black-box access* to $T_1(BA\mathbf{x}), \ldots, T_s(BA\mathbf{x})$ by exploiting their variable disjointness. From black-box access to $T_i(BA\mathbf{x})$, we get black-box access to $Q_{i,1}(BA\mathbf{x}), \ldots, Q_{i,m_i}(BA\mathbf{x})$, where $Q_{i,1}, \ldots, Q_{i,m_i}$ are the irreducible factors of T_i. Claim 2.3 then lets us find a $C \in \mathrm{GL}(n, \mathbb{F})$ such that $Q_{i,1}(BAC\mathbf{x}), \ldots, Q_{i,m_i}(BAC\mathbf{x})$, for all $i \in [s]$, are variable disjoint. At this point, we plan to recurse on $Q_{i,1}(BAC\mathbf{x}), \ldots, Q_{i,m_i}(BAC\mathbf{x})$ that are in the orbits of variable disjoint $+$-rooted ROFs of smaller size and depth.

Although the method looks promising, there are a few significant hurdles that render the basic approach almost useless. First, we shall see (in the next section) that the number of essential variables of the Hessian determinant of a canonical ROF can be dramatically smaller than n, although the ROF itself has no redundant variable. This is indeed a serious problem for the approach as the step of making $T_1(BA\mathbf{x}), \ldots, T_s(BA\mathbf{x})$ variable disjoint may break down completely. Second, even if we manage to make $T_1(BA\mathbf{x}), \ldots, T_s(BA\mathbf{x})$ variable disjoint, the complexity of the recursive algorithm may grow *exponentially* with the product depth of the ROF unless we generate super-efficient black-box access to $T_1(BA\mathbf{x}), \ldots, T_s(BA\mathbf{x})$. In the next section, we elaborate on these (and more) hurdles and explain how we overcome them and salvage the basic approach.

1.3.2 Outline of the ROF equivalence test: Salvaging the basic approach

Without loss of generality, assume that the root node of an ROF $\mathtt{C}$ is a $+$-gate; if not, use black-box polynomial factorization to reduce to the $+$-rooted case. We need to answer two questions:

(A) How do we efficiently find a transformation that makes the terms variable disjoint?

(B) How do we get efficient black-box access to the terms once they are variable disjoint?

We now elaborate on the technical hurdles that we encounter and deal with while answering these.

A. Making the terms variable disjoint

We know that the terms can be made variable disjoint if the number of essential variables in $\det(H_{\mathtt{C}})$ is exactly n. But this need not be the case. In fact, we face an even more basic hurdle.

- Hurdle 1: The Hessian determinant of a non-zero canonical ROF can be identically zero.

For instance, the Hessian determinant of $(x_1x_2 + x_3x_4)(x_5x_6 + x_7x_8) + (y_1y_2 + y_3y_4)(y_5y_6 + y_7y_8)$ is identically zero over $\mathbb{F}_3$; the Hessian determinant of $x_1x_2x_3 + x_4$ is zero over any $\mathbb{F}$. None of these ROFs have redundant variables (see Observation 2.7), and yet their Hessian determinants are zero.

When is the Hessian determinant non-zero? We show in Lemma 3.1 that the Hessian determinant of a non-zero n-variate canonical ROF $\mathtt{C}$ is non-zero provided $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n$ and none of the children of the top $+$-gate of $\mathtt{C}$ is a variable[15]. Henceforth, we assume that $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n$. We call a variable that is directly connected to a $+$-gate a *dangling variable*, and a variable that is directly connected to the top $+$-gate the *top dangling variable*. Since $\mathtt{C}$ is in canonical form (see Definition 2.6), it can have at most one top dangling variable.

[15] Observe that if the top $+$-gate has a variable child, then the Hessian determinant is identically zero over any $\mathbb{F}$.

Few words on the proof of Lemma 3.1: We show that the coefficient of a certain high degree monomial in $\det(H_{\mathtt{C}})$ is a product of "small", non-zero numbers. Depending on the structure of $\mathtt{C}$, we first carefully pick a variable x in it and treat $\det(H_{\mathtt{C}})$ as a univariate polynomial over $\mathbb{F}[\mathbf{x} \setminus \{x\}]$. We then show that the coefficient of the highest degree term in x is a product of the Hessian determinants of "smaller", $\times$-rooted ROFs. We repeat this process inductively on these smaller ROFs to show that their Hessian determinants are non-zero. The inductive process constructs a high degree monomial implicitly. The base case of the induction deals with Hessians of monomials of degree at least 2. The Hessian determinant of a degree d monomial is itself a monomial with a non-zero coefficient. Thus, the coefficient of the special monomial constructed by the inductive process is a product of the coefficients of the Hessian determinants of monomials.

The presence of a top dangling variable makes $\det(H_{\mathtt{C}})$ zero. We will see shortly how to prevent $\det(H_{\mathtt{C}})$ from vanishing. At first, let us assume that $\mathtt{C}$ has *no* top dangling variable. Now, even if the Hessian determinant of $\mathtt{C}$ is non-zero, there is no guarantee that the number of essential variables of $\det(H_{\mathtt{C}})$ is the maximum possible. This poses the second and the main hurdle.

- Hurdle 2: The number of essential variables of $\det(H_{\mathtt{C}}) \neq 0$ can be much smaller than n.

For example, the Hessian determinant of $x_1(x_2x_3 + x_4) + y_1(y_2y_3 + y_4)$ has merely two essential variables; the Hessian determinant of $x_1x_2x_3 + x_4x_5$ has only three essential variables. For ease of explanation, we split the above hurdle into two questions. The first one is,

- Hurdle 2a: Which variables of a canonical ROF $\mathtt{C}$ are essential for its Hessian determinant?

The notion of "skewed paths" turns out to be quite useful in answering this question.

Skewed paths, truly essential variables, and good and bad terms: A *skewed path* in $\mathtt{C}$ is a special structure that can be identified with a unique "marker" monomial (see Definition 3.1). In Claim 3.2, we show that every variable other than the dangling variables along skewed paths, the variables in quadratic forms along skewed paths, and the variables in the (top) quadratic form of $\mathtt{C}$, are *truly essential* for $\det(H_{\mathtt{C}})$ (see Definition 2.2)[16]. This knowledge enables us to categorize the terms of $\mathtt{C}$ into three types – *good*, *bad*, and the quadratic form of $\mathtt{C}$. A bad term looks like $x \cdot Q$, where $x \in \mathbf{x}$ and Q is a $+$-rooted ROF. In the example, both $x_1(x_2x_3 + x_4)$ and $y_1(y_2y_3 + y_4)$ are bad terms. In $x_1(x_2x_3 + x_4)$, the "marker" monomial x_1 (which is a variable in this simple case) defines a skewed path, x_2 and x_3 are the variables of the quadratic form along this skewed path, and x_4 is the dangling variable along this skewed path. Terms that are not bad and have degree ≥ 3 are good.

Making good terms variable disjoint. If T is a good (similarly, bad) term of $\mathtt{C}$, then we say $T(B\mathbf{x})$ is a good (respectively, bad) term of the input $f = \mathtt{C}(B\mathbf{x})$. It follows from Definition 3.1 that the skewed paths in $\mathtt{C}$ occur only in the bad terms of $\mathtt{C}$, and so, from Claim 3.2, all the variables of the good terms of $\mathtt{C}$ are truly essential for $\det(H_{\mathtt{C}})$. This fact along with Claim 2.4 and Observation 2.2 help us infer that a slight variant of the basic strategy given in Section 1.3.1 succeeds in finding an $A_0 \in \mathrm{GL}(n, \mathbb{F})$ such that the good terms of f become variable disjoint under the action of A_0. See Step 1 in Section 4.1 and Appendix G for a more detailed and pictorial overview of this step.

[16] See the paragraph before Claim 3.2 for relevant terminologies. Partitioning a set of essential variables into truly essential variables and ordinary essential variables helps us crucially in the arguments.

Making the good terms of f variable disjoint is the first step towards overcoming Hurdle 2. But it is far from sufficient even if $\mathtt{C}$ is devoid of bad terms, the top quadratic form, and the top dangling variable. This is because the algorithm may encounter bad terms, quadratic forms and dangling variables at deeper levels of the recursion, whence the basic strategy will fail. Therefore, we must answer the following (second) question to tackle the acute loss of essential variables in the Hessian determinant due to the presence of skewed paths in bad terms.

- Hurdle 2b: How do we handle the bad terms and the quadratic form of $\mathtt{C}$?

We call the dangling variables along skewed paths, the variables in quadratic forms along skewed paths, and the variables of the top quadratic form of $\mathtt{C}$ the *bad variables* of $\mathtt{C}$. The remaining variables are the *good variables*. Note that a good term of $\mathtt{C}$ has only good variables, whereas a bad term has both good and bad variables. For example, x_1 is a good variable of the bad term $x_1(x_2x_3 + x_4)$, and x_2, x_3, x_4 are its bad variables. By Claim 3.2, the good variables are truly essential for $\det(H_{\mathtt{C}})$, but they need not be the only essential variables. Some bad variables can be truly or ordinarily essential for $\det(H_{\mathtt{C}})$ or totally absent from $\det(H_{\mathtt{C}})$; this complicates the matter a bit.

Making the bad terms and the top quadratic form variable disjoint. It turns out that Claim 3.2, Claim 2.4 and Observation 2.2 together imply that the transformation A_0 is such that BA_0 maps every good variable of a (good or bad) term T_k to a linear form in $\mathbf{z}_k \subseteq \mathbf{x}$, where the variable sets $\mathbf{z}_k$ (as T_k runs over all good and bad terms) are disjoint. Let $\mathbf{z}$ be the disjoint union of these sets $\mathbf{z}_k$, and $\mathbf{y} := \mathbf{x} \setminus \mathbf{z}$. Let $\ell_x := BA_0 \circ x$ for $x \in \mathbf{x}$. Observe that the $\mathbf{y}$-variables appear only in the linear forms ℓ_x where x is a bad variable. Let $[\ell_x]_{\mathbf{y}}$ be ℓ_x restricted to the $\mathbf{y}$-variables. Loosely speaking, we make the bad terms and the top quadratic form of f variable disjoint in three (implicit) steps: "access" the linear forms $[\ell_x]_{\mathbf{y}}$, map them to distinct $\mathbf{y}$-variables, and then remove "external variables" from each of the terms. Let us elaborate on these steps by focusing on the bad terms.

Mapping "garbled" skewed paths back to monomials to access $[\ell_x]_{\mathbf{y}}$: How do we access $[\ell_x]_{\mathbf{y}}$, where x is a variable in a quadratic form along a skewed path or a dangling variable along a skewed path? The answer lies in the fact that a skewed path is identified with a unique "marker" monomial μ. This monomial can potentially help us access $[\ell_x]_{\mathbf{y}}$, where x a quadratic form or a dangling variable along the skewed path μ. But the problem is that the transformation BA_0 may have "garbled" the variables of μ. If for every variable z of μ, we find $\ell_z \in \mathbb{F}[\mathbf{z}]$, then we can map ℓ_z to a distinct $\mathbf{z}$-variable and get back a marker monomial – this works as z is a good variable. By Claim 3.1, such an ℓ_z is a factor of $\det(H_f)(A_0\mathbf{x})$. We can factorize $\det(H_f)(A_0\mathbf{x})$ and try to find ℓ_z, but there is a problem: $\det(H_f)(A_0\mathbf{x})$ might have other spurious linear factors that are not ℓ_z for any $z \in \mathbf{x}$. Fortunately, we can distinguish ℓ_z from spurious linear factors of $\det(H_f)(A_0\mathbf{x})$ by examining the number of essential variables of $f(A_0\mathbf{x})$ modulo affine forms; this crucial result is proved in Claim 2.1. So, we can safely assume without any loss of generality that $\ell_z = z$ for every variable z in μ.

Processing quadratic forms along skewed paths and the top quadratic form: We focus on a quadratic form $q = y_1y_2 + \ldots + y_{l-1}y_l$ along a skewed path μ, and let $\widetilde{q} = [\ell_{y_1}]_{\mathbf{y}}[\ell_{y_2}]_{\mathbf{y}} + \ldots + [\ell_{y_{l-1}}]_{\mathbf{y}}[\ell_{y_l}]_{\mathbf{y}}$. We can access $\widetilde{q}$ as follows: Treat $f(BA_0\mathbf{x})$ as a polynomial in $\mathbf{y}$ over $\mathbb{F}[\mathbf{z}]$ and extract out black-box access to the homogeneous degree-2 component in $\mathbf{y}$; call it $\widehat{q}$. As the degree-2 monomials in $\mathbf{y}$ are contributed only by the quadratic forms on skewed paths and the quadratic form of $\mathtt{C}$, and there are at most n different skewed paths, $\widehat{q}$ is n^3-sparse as a polynomial in $\mathbb{F}[\mathbf{y}, \mathbf{z}]$. We find the dense representation of $\widehat{q}$ using the sparse polynomial interpolation algorithm of [KS01]. Observe that the coefficient of μ in $\widehat{q}$ (as a polynomial in $\mathbf{z}$ over $\mathbb{F}[\mathbf{y}]$) is $\widetilde{q}$. Once we collect *all* the $\widetilde{q}$ for quadratic

forms along skewed paths, we map them simultaneously to quadratic SP polynomials in distinct $\mathbf{y}$-variables using Claim 2.3 and the QFE oracle. The existence of such a map A_1 is ensured by Claim 3.3 which shows that the variables of a quadratic form are either all truly essential for $\det(H_{\mathtt{C}})$ or they are absent from $\det(H_{\mathtt{C}})$. We then argue (in Claim 4.5) that $q(BA_0A_1\mathbf{x})$ can be expressed as $(y_1+h_1)(y_2+h_2)+\cdots+(y_{l-1}+h_{l-1})(y_l+h_l)$ for some (hitherto unknown) linear forms $h_1,\ldots,h_l \in \mathbb{F}[\mathbf{z}]$. A similar process for $\mu = 1$ takes care of the top quadratic form of $\mathtt{C}$. See Step 2.1 in Section 4.1 and Appendix G for a more detailed and pictorial overview of this step.

Handling dangling variables on skewed paths: Now let $\ell_x := BA_0A_1 \circ x$ and $\mathbf{u}$ be the $\mathbf{y}$-variables that have not been 'used up' by the QFE oracle in the previous step. Consider a dangling variable x along a skewed path μ. We can access $[\ell_x]_{\mathbf{u}}$ using μ, just as we accessed $\widetilde{q}$ before, by treating $f(BA_0A_1\mathbf{x})$ as a polynomial in $\mathbf{u}$ over $\mathbb{F}[\mathbf{x}\setminus\mathbf{u}]$ and extracting out the homogeneous degree-1 component in $\mathbf{u}$. However, unlike the variables of a quadratic form along a skewed path, a dangling variable x might not enjoy the property that it is either truly essential for $\det(H_{\mathtt{C}})$ or it is absent from $\det(H_{\mathtt{C}})$. So it may not be possible to map all the linear forms $[\ell_x]_{\mathbf{u}}$ for dangling variables along skewed paths to distinct $\mathbf{u}$-variables. This makes the argument here a bit subtle: We show, using Observation 4.2, Claim 4.7 and Observation 4.4, that it is sufficient to work with *any* basis $\mathscr{B}$ of the vector space spanned by the linear forms $[\ell_x]_{\mathbf{u}}$ as x varies over dangling variables along skewed paths. Mapping the elements of $\mathscr{B}$ to distinct $\mathbf{u}$-variables, using a transformation A_2, automatically 'takes care of' the linear forms outside $\mathscr{B}$. At a high level, this strategy works because the elements of $\mathscr{B}$ essentially corresponds to a set of redundant variables of $\det(H_{\mathtt{C}})$. See Step 2.2 in Section 4.1 and Appendix G for a more detailed and pictorial overview of this step.

Removing external variables from the terms: Let $\ell_x := BA_0A_1A_2 \circ x$ for $x \in \mathbf{x}$. For a bad term T_k, let $\mathbf{y}_k$ be the union of the $\mathbf{y}$-variables appearing in all ℓ_x, where x is a variable of a quadratic form along a skewed path in T_k, and the $\mathbf{u}$-variables present in all ℓ_x, where x is a dangling variable along a skewed path in T_k. The variables not in $\mathbf{z}_k \uplus \mathbf{y}_k$ are the *external variables* of T_k. Observe that the external variables appear in ℓ_x only if x is a dangling variable along a skewed path in T_k or a variable of a quadratic form along a skewed path in T_k. In this step, we intend to remove these external variables and complete the process of making the bad terms and the top quadratic form of f variable disjoint. At a high level, this is done by examining some carefully chosen first-order partials of $f(BA_0A_1A_2\mathbf{x})$ and engaging the skewed paths again to access the external variables. The proof of correctness of this step involves a few "disambiguation arguments" (see Observations 4.5, 4.6 and 4.7) which ensure that relevant monomials are generated "uniquely". See Step 2.3 in Section 4.1 and Appendix G for a more detailed and pictorial overview of this step.

Handling the top dangling variable. If $\det(H_{\mathtt{C}}) = 0$ (which, by Lemma 3.1, happens if and only if $\mathtt{C}$ has a top dangling variable) then we can reduce to the non-zero Hessian determinant case as follows: Apply a random transformation on the variable set $\mathbf{x} = \{x_1,\ldots,x_n\}$ and consider the Hessian of the resulting f with respect to only $x_1,\ldots,x_{n-1}$. Intuitively, the random transformation lets us assume two facts – one, the top dangling variable of $\mathtt{C}$ is x_n, i.e., $\mathtt{C} = \mathtt{C}_1(x_1,\ldots,x_{n-1}) + x_n$, where $\mathtt{C}_1$ is a canonical ROF with no top dangling variable; two, $f = \mathtt{C}_1(B\mathbf{x}) + \ell(\mathbf{x})$ for some $B \in \mathrm{GL}(n,\mathbb{F})$ such that $B \circ x_n = x_n$ and ℓ is an affine form. Now observe that the determinant of the Hessian of f with respect to $x_1,\ldots,x_{n-1}$ is an $\mathbb{F}$-multiple of $\det(H_{\mathtt{C}_1})(B\mathbf{x})$, which is non-zero as $\mathtt{C}_1$ has no top dangling variable. We can then remove the redundant variable x_n from $\det(H_{\mathtt{C}_1})(B\mathbf{x})$ and hope to find a $D \in \mathrm{GL}(n,\mathbb{F})$ such that $T_1(BD\mathbf{x}),\ldots,T_{s-1}(BD\mathbf{x})$ are variable disjoint, where $T_1,\ldots,T_{s-1}$ are the terms of $\mathtt{C}_1$. Once D is obtained, we are left with finding $\ell(D\mathbf{x})$

from black-box access to $f(D\mathbf{x}) = \mathtt{C}_1(BD\mathbf{x}) + \ell(D\mathbf{x})$. Indeed, the knowledge of D and $\ell(D\mathbf{x})$ is sufficient to construct an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $T_1(BA\mathbf{x}), \ldots, T_s(BA\mathbf{x})$ are variable disjoint (here, $T_s = x_n$). See Step 3 in Section 4.1 for a more detailed overview on how to find $\ell(D\mathbf{x})$ by exploiting the Hessian determinant again! A special case of this problem when ℓ is a constant also arises in the resolution of the final hurdle (stated below). We give the proof idea for this special case next.

B. Obtaining efficient black-box access to the terms

The above process finds an $A \in \mathrm{GL}(n, \mathbb{F})$ such that the terms $T_1(BA\mathbf{x}), \ldots, T_s(BA\mathbf{x})$ are variable disjoint. Let $r_i(\mathbf{x}_i) := T_i(BA\mathbf{x})$.

- Hurdle 3: How do we get *efficient* black-box access to $r_1(\mathbf{x}_1), \ldots, r_s(\mathbf{x}_s)$?

In other words, how do we simulate a black-box query to $r_i(\mathbf{x}_i)$ using *only one* query to the black-box for the input polynomial f. It is important to use only one query to f, as otherwise the time complexity of the recursive algorithm will become exponential in the product-depth of the ROF. The product depth of an n-variate ROF can be as high as $\Omega(n)$. We address this issue as follows.

At first, we examine the second-order derivatives of f to learn the variable sets $\mathbf{x}_1, \ldots, \mathbf{x}_s$ (see Claim 4.13). Then, we set the variables in $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_s$ to arbitrary field constants to reduce the problem to securing black-box access to $r_i(\mathbf{x}_i)$ from black-box access to $g_i := r_i(\mathbf{x}_i) + c$, where $c \in \mathbb{F}$ is unknown. If r_i is quadratic or linear, then we simply interpolate g_i and know r_i. Otherwise, we can still hope to learn c as it is the *unique* constant such that $g_i - c$ is reducible [17]. The uniqueness of c follows from the irreducibility of a $+$-rooted ROF (see Fact 2.5). But how do we learn c efficiently? The Hessian determinant comes in handy again.

Finding c. Suppose $r_i(\mathbf{x}_i) = r_{i,1}(\mathbf{x}_i) \cdot \ldots \cdot r_{i,m_i}(\mathbf{x}_i)$, where $r_{i,1}, \ldots, r_{i,m_i}$ are the irreducible factors of r_i, and $\deg(r_i) \geq 3$. It follows from Corollary 3.1 that $\det(H_{r_i})$, which equals $\det(H_{g_i})$, has as one of its irreducible factors an $\mathbb{F}$-multiple of $r_{i,j}$ for some $j \in [m_i]$. The efficient black-box polynomial factorization algorithm [KT90] gives us black-box access to all the irreducible factors of $\det(H_{g_i})$. Now suppose we pick the irreducible factor $\alpha \cdot r_{i,j}$, where $\alpha \in \mathbb{F}^\times$, from among the irreducible factors of $\det(H_{g_i})$. Define a random substitution map π on the variables of $\mathbf{x}_i$ as follows: $\pi(x) := c_x t$, where $c_x \in_r \mathbb{F}$ and t is a fresh variable, for every $x \in \mathbf{x}_i$. Interpolate the univariate polynomials $\pi(g_i)(t)$ and $\pi(\alpha \cdot r_{i,j})(t)$ that are non-constant with high probability, if $|\mathbb{F}|$ is sufficiently large. The degrees of $\pi(g_i)$ and $\pi(\alpha \cdot r_{i,j})$ are upper bounded by n. To find c, we set up and solve a linear system via the equation $\pi(g_i) = (a_{n-1}t^{n-1} + \ldots + a_0) \cdot \pi(\alpha \cdot r_{i,j}) + c_0$, [18] by pretending that $a_{n-1}, \ldots, a_0$ and c_0 are variables. The system has a solution that is obtained by choosing $a_{n-1}t^{n-1} + \ldots + a_0 = \pi(\alpha^{-1} \cdot \prod_{l \in [m_i] \setminus \{j\}} r_{i,l})$ and $c_0 = c$. This solution is unique. To see this, suppose $a_{n-1,1}, \ldots, a_{0,1}, c_{0,1}$ and $a_{n-1,2}, \ldots, a_{0,2}, c_{0,2}$ are two different solutions. Then,

$$((a_{n-1,1} - a_{n-1,2})t^{n-1} + \ldots + (a_{0,1} - a_{0,2})) \cdot \pi(\alpha \cdot r_{i,j}) + (c_{0,1} - c_{0,2}) = 0,$$

indicating that $\pi(\alpha \cdot r_{i,j})$ divides $(c_{0,1} - c_{0,2})$. But this is not possible as $\pi(\alpha \cdot r_{i,j})$ is not a constant. So, we solve the above system and declare the solution for c_0 as c. This procedure works if we pick an irreducible factor of $\det(H_{g_i})$ that is an $\mathbb{F}$-multiple of $r_{i,j}$ for some $j \in [m_i]$. But what if we

[17] More generally, this is true if r_i is a multilinear polynomial having at least two non-trivial factors. But, c need not be unique if r_i is not multilinear. For example, if $r_i = x^2$, then $g_i - c$ is reducible for both $c = 0$ and $c = 1$.

[18] As $\deg(\alpha \cdot r_{i,j}) \geq 1$, the degree of $\alpha^{-1} \cdot \prod_{l \in [m_i] \setminus \{j\}} r_{i,l}$ is at most $n - 1$.

pick a "wrong" factor? Indeed, the Hessian determinant can have other "spurious" factors. The point is that irrespective of what factor we choose, we can run the above procedure and find some c_0. If no c_0 is found, then we know immediately that a wrong factor is chosen. Otherwise, we check if $g_i - c_0$ is reducible, and if so, then take c_0 as c. The uniqueness of c implies that we always find the right c. Once we know c, we can simulate a black-box query to r_i using only one query to f.

Preparing for recursion. From efficient black-box access to r_i, we need to gain efficient black-box access to the irreducible factors of r_i as the algorithm essentially recurses on these factors. This is done as follows: Use the efficient black-box polynomial factorization algorithm [KT90] to get (not necessarily efficient) black-box access to $\alpha_j \cdot r_{i,j}$ for every $j \in [m_i]$, where $\alpha_j \in \mathbb{F}^\times$ and $\alpha_1 \cdot \alpha_2 \cdot \ldots \cdot \alpha_{m_i} = 1$. Claim 2.3 then allows us to find a $C_i \in \mathrm{GL}(|\mathbf{x}_i|, \mathbb{F})$ such that $\alpha_1 \cdot r_{i,1}(C_i\mathbf{x}_i), \ldots, \alpha_{m_i} \cdot r_{i,m_i}(C_i\mathbf{x}_i)$ are variable disjoint. Notice that we can easily get efficient black-box access to $r_i(C_i\mathbf{x}_i)$ from the efficient black-box for r_i. It is now sufficient to create an efficient black-box for $\alpha_j \cdot r_{i,j}(C_i\mathbf{x}_i)$ from the black-box for $r_i(C_i\mathbf{x}_i)$. Substitute the variables in $\alpha_l \cdot r_{i,l}(C_i\mathbf{x}_i)$ by random field constants for every $l \in [m_i]\backslash\{j\}$; denote this substitution map by ρ. Let $\beta_l = \rho(\alpha_l \cdot r_{i,l}(C_i\mathbf{x}_i))$. Observe that we know β_l from the already acquired (possibly inefficient) black-box for $\alpha_l \cdot r_{i,l}(C_i\mathbf{x}_i)$. Also, $\beta_l \neq 0$ with high probability. Then, the relation $\alpha_j \cdot r_{i,j}(C_i\mathbf{x}_i) = \rho(r_i(C_i\mathbf{x}_i)) \cdot \prod_{l\in[m_i]\backslash\{j\}} \beta_l^{-1}$ produces an efficient black-box for $\alpha_j \cdot r_{i,j}(C_i\mathbf{x}_i)$. The algorithm recurses on $\alpha_j \cdot r_{i,j}(C_i\mathbf{x}_i)$ with this black-box.

To summarize, irrespective of the level of the recursion, a required black-box can be obtained as an expression $\alpha f(C\mathbf{x} + \mathbf{c}) + \beta$, where $C \in M(n, \mathbb{F})$, $\mathbf{c} \in \mathbb{F}^n$, and $\alpha, \beta \in \mathbb{F}$ are known. Thus, the black-box query time is independent of the recursion depth. Moreover, the time taken to prepare a black-box for a subsequent level of the recursion (i.e., to make ready the knowledge of a relevant affine projection $C, \mathbf{c}$ and appropriate constants α, β) is independent of the recursion depth.

1.3.3 The approach for PE for orbits of ROFs

Let $f_1, f_2 \in \mathbb{F}[\mathbf{x}]$ be polynomials in the orbits of additive-constant-free ROFs. We can assume that they are equivalent. If not, then we run the algorithm on input f_1 and f_2, obtain an $A \in \mathrm{GL}(n, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^n$ and check if $f_1 = f_2(A\mathbf{x} + \mathbf{b})$ using the Schwartz-Zippel lemma [Sch80, Zip79]. This check will fail with high probability and we will conclude that f_1 and f_2 are not equivalent.

If $f_1 \sim f_2$, there exists a additive-constant-free ROF $\mathtt{C}$ such that $f_1, f_2 \in \mathrm{orb}(\mathtt{C})$. Using Theorem 1, we find $A_1, A_2 \in \mathrm{GL}(n, \mathbb{F})$ such that there exist permutation matrices $P_1, P_2 \in M(n, \mathbb{F})$, scaling matrices $S_1, S_2 \in M(n, \mathbb{F})$, and translation vectors $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{F}^n$ satisfying $f_1(A_1\mathbf{x}) = \mathtt{C}(P_1S_1\mathbf{x} + \mathbf{d}_1)$ and $f_2(A_2\mathbf{x}) = \mathtt{C}(P_2S_2\mathbf{x} + \mathbf{d}_2)$. We reconstruct the ROFs $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$ that are in the PS orbit (see Definition 2.5) of $\mathtt{C}$. It turns out that an ROF in the PS orbit of a canonical ROF can be reconstructed *uniquely* up to scaling of the leaves (see Section F). As $\mathtt{C}$ is additive-constant-free, all leaves of $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$ are either variables, or constants that act as translation or scaling of the variables. Let $\mathtt{C}_1$ and $\mathtt{C}_2$ be the ROFs obtained by reconstructing $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$ respectively. Once we recover the scaling and the translation of all the variables in $\mathtt{C}_1$ and $\mathtt{C}_2$ (see Section F.3), we can transform $\mathtt{C}_1$ and $\mathtt{C}_2$ in such a way that they resemble $\mathtt{C}$ and differ by only a permutation of the variables. We find this permutation using the tree isomorphism algorithm in [AHU83] (see Fact A.4). Then, an $A \in \mathrm{GL}(n, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^n$ such that $f_1 = f_2(A\mathbf{x} + \mathbf{b})$ can be found using this isomorphism, the recovered scaling and translation, and A_1, A_2.

We believe that the additive-constant-free restriction can be completely removed with some more technical effort. As an evidence supporting this belief, we show in Section E that the additive-constant-free restriction can be dispensed with for depth-4 ROFs.

1.3.4 Brief comparison with other approaches

The Lie algebra associated with a polynomial[19] has been used extensively to design equivalence tests for a number of important polynomial families [Kay12,Gro12,KNST19,GS19,GGKS19,MNS20]. Can Lie algebras be used for ROF equivalence tests? It turns out that the Lie algebra of the sum-product polynomial SP is sufficiently rich and one can indeed devise an equivalence test for SP based on it (provided $\deg(\mathsf{SP}) \geq 3$). But this is not the case for an arbitrary ROF. For example, the Lie algebra of the ROF $(x_1+1)(x_2+2)(x_3+3)+(x_4+4)(x_5+5)(x_6+6)$ is trivial; it gives practically no information about the ROF.

It was observed in [MS21] that the average-case ANF reconstruction algorithm from [GKQ14] already gives an equivalence test for ROANFs. But the time complexity of the algorithm in [GKQ14] degrades rapidly with increasing fan-in of the $+$-gates. An arbitrary ROF has no fan-in restriction. So, it is unclear if the algorithm in [GKQ14] can be easily adapted to give an ROF equivalence test while preserving its efficiency.

[MS21] also observed that the average-case homogeneous depth-3 circuit reconstruction algorithm in [KS19] gives an equivalence test for the sum-product polynomial[20]. The approach in [KS19, GKS20] is based on a scheme for obtaining an average-case learning algorithm for a circuit class from lower bounds for the same class. The scheme is potentially useful in developing an average-case formula reconstruction algorithm from formula lower bounds. Average-case formula reconstruction being a (possibly) stronger objective, it is likely to imply an equivalence test for ROFs. But currently, no super-polynomial lower bound is known for formulas. So the approach in [KS19, GKS20] does not yet provide an equivalence test for general ROFs.

1.4 Related work

Cubic form equivalence and associated problems. It has long been known that the polynomial equivalence problem can be solved efficiently for quadratic forms (see Section A.2). But the difficulty of the problem increases sharply for cubic forms. Cubic form equivalence is at least as hard as the graph isomorphism problem and possibly harder [AS05]. Several other important problems–algebra isomorphism, matrix space isometry, matrix space conjugacy, group isomorphism for certain p-groups, 3-tensor isomorphism, and trilinear form equivalence–are polynomial-time equivalent to cubic form equivalence [GQ21, FGS18, BW15, AS06, AS05]. The best-known worst-case complexity of cubic form equivalence is not significantly better than that of polynomial solvability. However, a moderately exponential-time algorithm is known for a natural average-case version of the cubic form equivalence problem over finite fields [GQT21].

The supposed hardness of constant-degree form equivalence (even in the average case) led to the development of a cryptographic authentication scheme [Pat96]. The main problem studied in this context is known as *isomorphism of polynomials with one secret* (IP1S). In the IP1S problem, we are given two tuples of polynomials $(f_1,\ldots,f_m)$ and $(g_1,\ldots,g_m)$ and we wish to check if there is an invertible A such that $f_i = g_i(A\mathbf{x})$ for all $i \in [m]$. Efficient algorithms are known only for the quadratic IP1S problem [IQ19, BFP15], i.e., when $f_1,\ldots,f_m$ and $g_1,\ldots,g_m$ are quadratic forms[21].

[19] The Lie algebra associated with (the group of symmetries of) an n-variate polynomial f is the space of matrices $(a_{i,j})_{i,j\in[n]}$ satisfying the equation $\sum_{i,j\in[n]} a_{i,j}\cdot x_j\frac{\partial f}{\partial x_i} = 0$.

[20] The SP polynomial is thus a unique example for which three different equivalence testing algorithms are known.

[21] Even the quadratic case is quite non-trivial for IP1S as we are dealing with tuples of polynomials.

Special polynomial families. Spurred by applications in algebraic and geometric complexity theory, [Kay11] studied an interesting variant of the polynomial equivalence problem for well-known polynomial families. In this setting, we fix a polynomial family (say, the determinant), take input black-box access to a polynomial f, and check if f is in the orbit of a polynomial in the family. Starting with the algorithms in [Kay11], efficient equivalence tests were given for a number of polynomial families, namely the power symmetric polynomials, the elementary symmetric polynomials, the sum-product polynomials, the determinant, the permanent, the iterated matrix multiplication (IMM) polynomials, the design polynomials, and the continuant polynomials [Kay11, Kay12, Gro12, KNST19, GS19, GGKS19, MNS20, MS21]. Some of these equivalence tests have interesting applications in circuit reconstruction [KNS19, BGKS21]. [GKP18] gave an efficient equivalence test for the family of sums of univariates, which is a generalization of the family of power symmetric polynomials. Recently, [KS21b, KS21a] gave a randomized polynomial-time algorithm (in the Turing machine model) to solve the decision version of equivalence testing for the power symmetric polynomials over $\mathbb{C}$, where the input polynomial has rational coefficients.

Results on ROFs. Reconstruction algorithms and hitting-sets have been intensely studied for ROFs [HH91, BHH95a, BC98, BB98, SV14, Vol16, MV18]. Deterministic polynomial-time reconstruction and hitting-sets are known for ROFs [MV18, SV14]. Recently, [MS21, ST21] gave quasi-polynomial time hitting-sets for orbits of ROFs. It is worth noting that despite its apparent weakness, the ROF model has served as a testbed for developing effective tools and techniques for analyzing circuits. For example, a construction of k-independent polynomial maps that played a vital role in hitting-set constructions for several important circuit classes originated from a study of ROFs [SV14]. [RS11] studied a special case of ET for ROFs; they gave a polynomial-time algorithm to check if two constant-free ROFs are permutation equivalent[22]. They also observed that permutation equivalence for read-2 formulas and PE for read-4 formulas are graph isomorphism hard. Efficient learning algorithms are also known for Boolean read-once formulas [AHK93, BHH95b]. Read-once formulas have been used as "hard functions" in finer lower bounds and separation results for low-depth Boolean circuits [Sip83, HRST17, HHTT22]. [Gur77, KLN+93] gave a characterization of functions computable by Boolean ROFs, and [Vol16] gave a characterization of polynomials computable by arithmetic ROFs.

Other results on PE. Two natural special cases of the polynomial equivalence problem are translation equivalence and scaling equivalence. In the translation equivalence problem, we are given black-box access to two n-variate polynomials f and g and we wish to determine if there is a $\mathbf{b} \in \mathbb{F}^n$ such that $f = g(\mathbf{x} + \mathbf{b})$. In the scaling equivalence problem, we wish to find out if there a diagonal matrix $S \in \mathrm{GL}(n, \mathbb{F})$ such that $f = g(S\mathbf{x})$. Efficient algorithms are known for both translation and scaling equivalence tests [DdOS14, Kay12, BRS17].

[Kay11] gave an efficient algorithm to check if a given n-variate polynomial f is equivalent to a multilinear polynomial, provided the dimension of the space spanned by the second-order partials of f is $\binom{n}{2}$. This result implies that equivalence test is easy for random multilinear polynomials. But notice that the dimension of the second-order partials of a typical ROF (for e.g., the sum-product polynomial) is substantially smaller than $\binom{n}{2}$.

[22] Two polynomials $f, g \in \mathbb{F}[\mathbf{x}]$ are permutation equivalent if there is a permutation matrix P such that $f = g(P\mathbf{x})$.

2 Preliminaries

Notations. For $n \in \mathbb{N}, [n] = \{1, \ldots, n\}$ and $\mathbf{x} = \{x_1, \ldots, x_n\}$. For $g \in \mathbb{F}[\mathbf{x}]$, $\text{var}(g)$ is the set of variables appearing in g. The space of polynomials having degree at most d will be denoted as $\mathbb{F}[\mathbf{x}]_{\leq d}$. For $S \subseteq \mathbb{F}[\mathbf{x}]$, $\langle S \rangle$ is the $\mathbb{F}$-linear space spanned by S. $\text{GL}(n, \mathbb{F})$ is the set of $n \times n$ invertible matrices over $\mathbb{F}$. For $A \in \text{GL}(n, \mathbb{F})$, $A\mathbf{x}$ is the column vector obtained by multiplying A with $(x_1 \cdots x_n)^T$. A detailed list of other notations is given in Table 1 of the appendix.

2.1 Structural preliminaries

2.1.1 Essential and redundant variables

Definition 2.1 (Essential and redundant variables)**.** The number of *essential variables* of an n-variate $g \in \mathbb{F}[\mathbf{x}]$ is $s := \min_{A \in \text{GL}(n,\mathbb{F})} |\text{var}(g(A\mathbf{x}))|$. The number of *redundant variables* of g is $(n - s)$.

Following [Car06], we denote the number of essential variables of g by $N_{ess}(g)$. [Car06] gave a polynomial-time algorithm that takes input the coefficient vector of g and finds an $A \in \text{GL}(n, \mathbb{F})$ such that $|\text{var}(g(A\mathbf{x}))| = N_{ess}(g)$. [Kay11] gave a randomized polynomial-time algorithm that does the same given black-box access to g. These algorithms use a neat relation between $N_{ess}(g)$ and $\dim \left\langle \frac{\partial g}{\partial x} : x \in \mathbf{x} \right\rangle$. See Claim 2.3 in [KNST19] for a proof of the following fact.

Fact 2.1 (Essential variables and partials)**.** *Let $d \in \mathbb{N}$ and $\text{char}(\mathbb{F}) = 0$ or $> d$. If $g \in \mathbb{F}[\mathbf{x}]_{\leq d}$, then $N_{ess}(g) = \dim \left\langle \frac{\partial g}{\partial x} : x \in \mathbf{x} \right\rangle$. For $\mathbf{z} \subseteq \mathbf{x}$, $\left\{ \frac{\partial g}{\partial z} : z \in \mathbf{z} \right\}$ is a basis of $\left\langle \frac{\partial g}{\partial x} : x \in \mathbf{x} \right\rangle$ if and only if there is an $A \in \text{GL}(|\mathbf{x}|, \mathbb{F})$ that maps every variable in $\mathbf{x} \setminus \mathbf{z}$ to itself, $\text{var}(g(A\mathbf{x})) = \mathbf{z}$, and $N_{ess}(g(A\mathbf{x})) = |\mathbf{z}|$.*

We say a set $\mathbf{z} \subseteq \mathbf{x}$ is a *set of essential variables* of g if $\left\{ \frac{\partial g}{\partial z} : z \in \mathbf{z} \right\}$ is a basis of $\left\langle \frac{\partial g}{\partial x} : x \in \mathbf{x} \right\rangle$; variables in $\mathbf{x} \setminus \mathbf{z}$ are *redundant* for g. We categorize the essential variables further as follows.

Definition 2.2 (Truly and ordinary essential variables)**.** An $x \in \mathbf{x}$ is a *truly essential variable* of $g \in \mathbb{F}[\mathbf{x}]$ if for every $A \in \text{GL}(|\mathbf{x}|, \mathbb{F})$ that maps x to itself, $x \in \text{var}(g(A\mathbf{x}))$. If $\mathbf{z}$ is a set of essential variables of g, then a $z \in \mathbf{z}$ that is not truly essential is an *ordinary essential variable* of g in $\mathbf{z}$.

Observation 2.1 (Characterizing truly essential variables using partials)**.** *Let $d \in \mathbb{N}$ and $\text{char}(\mathbb{F}) = 0$ or $> d$. If $g \in \mathbb{F}[\mathbf{x}]_{\leq d}$, then $x \in \mathbf{x}$ is a truly essential variable of g if and only if $\sum_{x' \in \mathbf{x}} \alpha_{x'} \frac{\partial g}{\partial x'} = 0$ for $\alpha_{x'} \in \mathbb{F}$ implies $\alpha_x = 0$, i.e., no $\mathbb{F}$-linear dependence of $\left\{ \frac{\partial g}{\partial x'} : x' \in \mathbf{x} \right\}$ involves $\frac{\partial g}{\partial x}$.*

It follows that every set of essential variables of g contains all the truly essential variables. The proof of the observation is in Section B.1. The next fact follows from the proof of Fact 2.1.

Fact 2.2 (Structure of a matrix for removing redundant variables)**.** *Let $d \in \mathbb{N}, \text{char}(\mathbb{F}) = 0$ or $> d$, and $g \in \mathbb{F}[\mathbf{x}]_{\leq d}$. Let $\mathbf{z}$ be a set of essential variables of g, $\mathbf{z}_1$ the set of truly essential variables of g, $\mathbf{z}_2 = \mathbf{z} \setminus \mathbf{z}_1$, and $\mathbf{y} = \mathbf{x} \setminus \mathbf{z}$. Then, there is an $A \in \text{GL}(|\mathbf{x}|, \mathbb{F})$ that maps every variable in $\mathbf{z}_1 \uplus \mathbf{y}$ to itself, maps every $z \in \mathbf{z}_2$ to a linear form in $\mathbf{y} \uplus \{z\}$, and $\text{var}(g(A\mathbf{x})) = \mathbf{z}$.*

The proofs of the following observations are given in Sections B.2, B.3, and B.4.

Observation 2.2 (Truly essential variables map to linear forms in essential variables)**.** *Let $d \in \mathbb{N}, \text{char}(\mathbb{F}) = 0$ or $> d$, $\mathbf{x}$ and $\mathbf{y}$ be disjoint sets of variables, and $h \in \mathbb{F}[\mathbf{x}]_{\leq d}$. Let $\mathbf{z} \subseteq \mathbf{x} \uplus \mathbf{y}$ and $A \in \text{GL}(|\mathbf{x}| + |\mathbf{y}|, \mathbb{F})$ such that $|\mathbf{z}| = N_{ess}(h)$ and $h(A \cdot (\mathbf{x}, \mathbf{y})^T) \in \mathbb{F}[\mathbf{z}]$ (where we pretend that h is a polynomial in $\mathbf{x} \uplus \mathbf{y}$). Then, A maps every truly essential variable of h to a linear form in $\mathbf{z}$.*

Observation 2.3 (Truly essential variables from factors)**.** *Let $d \in \mathbb{N}, \mathrm{char}(\mathbb{F}) = 0$ or $> d$, and $\mathbf{x}$ and $\mathbf{y}$ be disjoint sets of variables. Let $h(\mathbf{x},\mathbf{y}) = g(\mathbf{x})^e \cdot p(\mathbf{x},\mathbf{y}) \in \mathbb{F}[\mathbf{x},\mathbf{y}]$, where $g(\mathbf{x}), p(\mathbf{x},\mathbf{y})$ are coprime, $\deg(h) \leq d$, and $e \geq 1$. If $N_{ess}(g) = |\mathbf{x}|$, then every $\mathbf{x}$-variable is truly essential for h.*

Observation 2.4 (Truly essential pairs of variables)**.** *Let $d \in \mathbb{N}, \mathrm{char}(\mathbb{F}) = 0$ or $> d$, and $\{x_1, x_2\}$ and $\mathbf{y}$ be disjoint sets of variables. Let $h(x_1, x_2, \mathbf{y}) = \sum_{i \geq 0} p_i(\mathbf{y}) \cdot (x_1 x_2)^i$ be a polynomial of degree at most d such that $p_i(\mathbf{y}) \neq 0$ for some $i \geq 1$. Then, x_1 and x_2 are truly essential for h.*

2.1.2 Essential variables modulo affine forms

Let $g \in \mathbb{F}[\mathbf{x}]$ and $\ell = \sum_{i \in [n]} \alpha_i x_i + \beta$, a non-constant affine form in $\mathbb{F}[\mathbf{x}]$. Let $I = \{i \in [n] : \alpha_i \neq 0\}$, and $\prec$ a monomial ordering on $\mathbb{F}[\mathbf{x}]$. Suppose, x_j has the highest precedence among $\{x_i : i \in I\}$ according to $\prec$. There is a natural ring isomorphism between the quotient ring $\mathbb{F}[\mathbf{x}]/\langle \ell \rangle$ and $\mathbb{F}[\mathbf{x} \setminus \{x_j\}]$, where $\langle \ell \rangle$ is the ideal generated by ℓ. We define g modulo ℓ (denoted g_ℓ) as:

$$g_\ell := g\left(x_1, \ldots, x_{j-1}, -{\alpha_j}^{-1}\left(\sum_{i \in [n]\setminus\{j\}} \alpha_i x_i + \beta\right), x_{j+1}, \ldots, x_n\right).$$

Definition 2.3. The number of essential variables of g modulo ℓ is defined as $N_{ess}(g_\ell)$.

Notice that the ordering $\prec$ is implicit in the above definition. But, the following observation shows that the exact choice of $\prec$ is unimportant here. Its proof is given in Section B.5.

Observation 2.5 (Soundness of Definition 2.3)**.** *For $j \in I$, let $W_j = \left\langle \frac{\partial \varphi_j(g)}{\partial x} : x \in \mathbf{x} \right\rangle$, where $\varphi_j(g)$ is obtained by substituting x_j in g by ${-\alpha_j}^{-1}\left(\sum_{i \in [n]\setminus\{j\}} \alpha_i x_i + \beta\right)$. Then, for $j_1, j_2 \in I$, $\dim W_{j_1} = \dim W_{j_2}$.*

2.1.3 Orbit of a polynomial

Definition 2.4 (Orbit of a polynomial)**.** The *orbit of* $g \in \mathbb{F}[\mathbf{x}]$, denoted $\mathrm{orb}(g)$, is defined as $\mathrm{orb}(g) := \{g(A\mathbf{x} + \mathbf{b}) : A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F}),\ \mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}\}$.

Remark. Our results, especially Theorem 1, hold even if we consider a more general definition of the orbit of a polynomial: Let $\mathbf{x}, \mathbf{y}$ be sets of n and m variables, where $n \geq m$, and $h \in \mathbb{F}[\mathbf{x}], g \in \mathbb{F}[\mathbf{y}]$. Then, $h \in \mathrm{orb}(g)$ if there is exist a $A \in \mathbb{F}^{m \times n}$ of rank m and $\mathbf{b} \in \mathbb{F}^m$ such that $h = g(A\mathbf{x} + \mathbf{b})$.

Definition 2.5 (PS orbit of a polynomial)**.** Let $g, h \in \mathbb{F}[\mathbf{x}]$. We say that h is in the *PS orbit of* g, denoted PS-orb(g), if there exist a $|\mathbf{x}| \times |\mathbf{x}|$ permutation matrix P, a $|\mathbf{x}| \times |\mathbf{x}|$ invertible diagonal (scaling) matrix S, and a $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$ such that $h = g(PS\mathbf{x} + \mathbf{b})$.

The following facts are easy to prove.

Fact 2.3. *If $h \in \mathrm{orb}(g)$, then $N_{ess}(g) = N_{ess}(h)$.*

Fact 2.4. *Let $d \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $> d$, $g \in \mathbb{F}[\mathbf{x}]_{\leq d}$, $\ell \in \mathbb{F}[\mathbf{x}]$ a non-constant affine form, $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$, $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$, $g' = g(A\mathbf{x} + \mathbf{b})$, and $\ell' = \ell(A\mathbf{x} + \mathbf{b})$. Then, $N_{ess}(g'_{\ell'}) = N_{ess}(g_\ell)$.*

2.1.4 Read-once formulas

An *arithmetic formula* $\mathtt{C}$ over $\mathbb{F}$ is a tree whose leaves are labelled by variables and $\mathbb{F}$-constants, other nodes (gates) are labelled by $+$ and $\times$ operations, and edges by $\mathbb{F}$-constants. A node v computes a polynomial naturally: if v is a leaf, it computes its label; if v is a $+$ gate (similarly, a $\times$ gate) having $v_1, \ldots, v_m$ as children such that for every $i \in [m]$, v_i computes $g_i \in \mathbb{F}[\mathbf{x}]$ and the edge connecting v and v_i is labelled by $\alpha_i \in \mathbb{F}$, then v computes $\sum_{i\in[m]} \alpha_i g_i$ (respectively, $\prod_{i\in[m]} \alpha_i g_i$). We will identify a node with the polynomial it computes and $\mathtt{C}$ with the polynomial computed by its root node. Without loss of generality, $\mathtt{C}$ has alternate layers of $+$ and $\times$ gates, every non-leaf gate has fan-in at least two, and every child of a $\times$ gate computes a non-constant polynomial.

An arithmetic formula $\mathtt{C}$ over $\mathbb{F}$ is a *read-once formula (ROF)* if every leaf in $\mathtt{C}$ is labelled by either a distinct variable or an $\mathbb{F}$-constant. The product-depth of $\mathtt{C}$, denoted Δ, is the number of $\times$ gates in a longest path in $\mathtt{C}$ from the root to a leaf. We call $\mathtt{C}$ a $+$-rooted (similarly, a $\times$-rooted) ROF if the root of $\mathtt{C}$ is a $+$ gate (respectively, a $\times$ gate). The following fact is easy to verify.

Fact 2.5 (Irreducibility of an ROF)**.** *The polynomial computed by a $+$-rooted ROF is irreducible over $\mathbb{F}$.*

Definition 2.6 (Canonical ROF)**.** An ROF $\mathtt{C}$ is *canonical* if it satisfies the following properties:

1. $\mathtt{C}$ has alternate layers of $+$ and $\times$ gates.
2. Every non-leaf gate in $\mathtt{C}$ has fan-in at least 2.
3. Every child of a $\times$ gate computes a non-constant polynomial.
4. There are no labels on the edges of $\mathtt{C}$.
5. A $+$ gate has at most one constant and at most one variable among its children, but not both.
6. Suppose there is a $+$ gate that has among its children a variable and a $\times$ gate v such that v has two children – a variable and a $+$ gate v'. Then, v' has no constant among its children.

Let $\mathtt{C}$ be a $+$-rooted canonical ROF over $\mathbb{F}$. The equation $\mathtt{C} = T_1 + \cdots + T_s + \gamma$ means that $T_1, \ldots, T_s$ are the non-constant children and $\gamma \in \mathbb{F}$ is the constant child of the root $+$ gate. Note that a constant in a canonical ROF $\mathtt{C}$ only appears as a child of a $+$ gate. Thus, all constants present in $\mathtt{C}$ are additive-constants. An example of a canonical ROF is an ROANF.

Definition 2.7 (ROANF)**.** A canonical ROF $\mathtt{C}$ is in the *read-once alternating normal form (ROANF)* if it is a complete binary tree, the root of $\mathtt{C}$ is a $+$ gate, the bottom-most layer of $\mathtt{C}$ contains $\times$ gates, and all the leaves are labelled with distinct variables.

Definition 2.8. An ROF is *additive-constant-free* if it has no additive-constants.

The following observations are proved in Section B.6 and B.7 respectively.

Observation 2.6 (Orbit of a canonical ROF)**.** *Let $\mathtt{C}$ be an ROF over $\mathbb{F}$. Then, there is a canonical ROF $\mathtt{C}'$ over $\mathbb{F}$ such that $\mathtt{C}' \in \mathrm{orb}(\mathtt{C})$. If $\mathtt{C}$ is additive-constant-free, then so is $\mathtt{C}'$.*

Observation 2.7 (Essential variables of a canonical ROF)**.** *The set of variables labelling the non-constant leaves of a canonical ROF $\mathtt{C}$ is the set of essential variables of $\mathtt{C}$, i.e., $\mathtt{C}$ has no redundant variable.*

If $\mathtt{C}$ is not canonical, then all the variables of $\mathtt{C}$ need not be essential. For e.g., $x_1 + \cdots + x_n$ is an ROF with only one essential variable. The above observations imply the following:

Observation 2.8. *Let $\mathtt{C}$ be an ROF, $\mathtt{C}'$ a canonical ROF, and $\mathtt{C}' \in \mathrm{orb}(\mathtt{C})$. Then, $N_{ess}(\mathtt{C}) = |\mathrm{var}(\mathtt{C}')|$.*

We now state an important property of a canonical ROF which will be used in the equivalence test in Section 4. A proof of this is given in Section B.8.

Claim 2.1 (Canonical ROF modulo an affine form)**.** *Let $n \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) \neq 2$, $|\mathbb{F}| > n$, $\mathtt{C}$ be a $+$-rooted n-variate canonical ROF, and ℓ an affine form which is not a constant multiple of some variable. Then, $N_{ess}(\mathtt{C}_\ell) \geq n - 2$.*

2.1.5 Hessian of a polynomial

Definition 2.9 (Hessian of a polynomial)**.** The *Hessian* of $g \in \mathbb{F}[\mathbf{x}]$, denoted as H_g, is the $n \times n$ matrix whose (i, j)-th entry is $\frac{\partial^2 g}{\partial x_i \partial x_j}$. The determinant of H_g is called the *Hessian determinant* of g.

The Hessian matrix appears naturally in the Taylor expansion of a polynomial and has important applications in optimization, second derivative tests, etc. In algebraic complexity, the rank of the Hessian plays a crucial role in the best known lower bound on the determinantal complexity of the permanent [MR04, CCL10]. As mentioned in Section 1, the Hessian determinant is an effective tool for designing equivalence tests for the sum-product polynomial, the power symmetric polynomial [Kay11], and the sum of univariates model [GKP18]. A suitable 4-th order generalization of the Hessian has been used in [GKP18] to study the Waring decomposition problem in the average case. In this work, we focus on understanding the essential variables of the Hessian determinant of a general ROF (see Section 3) to devise an equivalence test for ROFs.

A few basic properties of the Hessian are given below.

Observation 2.9. *Let $\mathtt{C} = T_1 + \cdots + T_s + \gamma$ be an ROF over $\mathbb{F}$, where for every $l \in [s]$, T_l is a $\times$-rooted sub-ROF of $\mathtt{C}$ and $\gamma \in \mathbb{F}$. Suppose the rows and columns of $H_{\mathtt{C}}$ are labelled by $\mathrm{var}(T_1), \ldots, \mathrm{var}(T_s)$ in order. Then, $H_{\mathtt{C}}$ is a block diagonal matrix, where for $l \in [s]$, the l-th block on the diagonal is H_{T_l}.*

Fact 2.6 (Chain rule)**.** *Let $g \in \mathbb{F}[\mathbf{x}]$ and $h = g(A\mathbf{x})$ for $A \in M(|\mathbf{x}|, \mathbb{F})$. Then, $H_h = A^T \cdot H_g(A\mathbf{x}) \cdot A$.*

Fact 2.7. *Let $g \in \mathbb{F}[\mathbf{x}]$ and $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$. Then, $H_{g(\mathbf{x}+\mathbf{b})} = H_g(\mathbf{x} + \mathbf{b})$.*

Fact 2.8. *Let $g \in \mathbb{F}[\mathbf{x}]$, $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$, $\mathbf{b} \in \mathbb{F}^{|\mathbf{x}|}$ and $h = g(A\mathbf{x} + \mathbf{b})$. Then, $\det(H_h) = \alpha^2 \cdot \det(H_g)(A\mathbf{x} + \mathbf{b})$, where $\alpha = \det(A)$.*

2.2 Algorithmic preliminaries

Algorithms to remove redundant variables from a polynomial are known [Car06, Kay11]. In the claim below, we mention a slightly general version of such an algorithm. Its proof follows from Fact 2.1 (see also the proof of Claim 2.3 in [KNST19]).

Claim 2.2 (Elimination of redundant variables)**.** *Let $d \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $> d$, and $|\mathbb{F}| \geq 2|\mathbf{x}|d$. There is a randomized $\mathrm{poly}(|\mathbf{x}|, d)$ time algorithm* Remove-Redundant-Vars *that takes input black-box access to a $g \in \mathbb{F}[\mathbf{x}]_{\leq d}$ and a set $\mathbf{y} \subseteq \mathbf{x}$ s.t. $\mathbf{x} \setminus \mathbf{y}$ contains a set of essential variables of g, and outputs an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ s.t. A maps every $\mathbf{y}$-variable to itself and $g(A\mathbf{x})$ is $\mathbf{y}$-free and has no redundant variables.*

If $g_1, \ldots, g_m$ are pairwise variable disjoint, then it can be shown that $N_{ess}(g_1 \cdots g_m) = N_{ess}(g_1) + \ldots + N_{ess}(g_m)$ over any field. The following claim proves the converse and, more importantly, provides an algorithm to find a transformation that makes $g_1, \ldots, g_m$ pairwise variable disjoint.

Claim 2.3 (Making polynomials variable disjoint)**.** *Let $d \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $> d$, and $|\mathbb{F}| \geq 2|\mathbf{x}|d$. There is a randomized $\mathrm{poly}(|\mathbf{x}|, d)$ time algorithm that takes input black-box access to $g_1, \ldots, g_m \in \mathbb{F}[\mathbf{x}]$, where $g_1 \cdots g_m \in \mathbb{F}[\mathbf{x}]_{\leq d}$ and $N_{ess}(g_1 \cdots g_m) = \sum_{i \in [m]} N_{ess}(g_i)$, and outputs an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ such that $g_1(A\mathbf{x}), \ldots, g_m(A\mathbf{x})$ are pairwise variable disjoint and individually free of redundant variables.*

The claim is proved in Section B.9. The next claim, which generalizes the above claim, is used crucially in the equivalence test presented in Section 4. It is proved in Section B.10.

Claim 2.4 (Making factors variable disjoint)**.** *Let $d \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $> d$, and $|\mathbb{F}| \geq \max\left\{2|\mathbf{x}|d, d^6\right\}$. There is a randomized $\mathrm{poly}(|\mathbf{x}|, d)$ time algorithm that takes input black-box access to an $f = g(B\mathbf{x} + \mathbf{d})$, where $B \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$, $\mathbf{d} \in \mathbb{F}^{|\mathbf{x}|}$, and $g \in \mathbb{F}[\mathbf{x}]_{\leq d}$ such that $g = g_1 \cdots g_m$ for pairwise variable disjoint $g_1, \ldots, g_m \in \mathbb{F}[\mathbf{x}]_{\leq d}$, and does the following: (Here, $B, \mathbf{d}, g$, and $g_1, \ldots, g_m$ are unknown to the algorithm.)*

1. *It computes an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ such that $g_1(BA\mathbf{x} + \mathbf{d}), \ldots, g_m(BA\mathbf{x} + \mathbf{d})$ are pairwise variable disjoint and individually free of redundant variables. ($g_1, \ldots, g_m$ need not be irreducible.)*
2. *It computes a set V of pairwise disjoint subsets of $\mathbf{x}$ such that for every $i \in [m]$, there exist $h_{i,1}, \ldots, h_{i,m_i} \in \mathbb{F}[\mathbf{x}]$ satisfying $\prod_{l \in [m_i]} h_{i,l} = g_i(B\mathbf{x} + \mathbf{d})$, and $V = \{\mathrm{var}(h_{i,l}(A\mathbf{x})) : i \in [m], l \in [m_i]\}$.*

3 The Hessian of an ROF

In this section, we state some important properties of the Hessian determinant of a canonical ROF. These properties play a crucial role in the equivalence test given in Section 4 and allow us to use the Hessian determinant to learn valuable information about the matrix mapping the input polynomial to a canonical ROF. We shall denote the Hessian of a polynomial g by H_g.

Lemma 3.1 ($\det(H_{\mathtt{C}}) \neq 0$)**.** *Let $n \in \mathbb{N}$ and $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n$. Let $\mathtt{C} = T_1 + \cdots + T_s + \gamma$ be a canonical ROF over $\mathbb{F}$, where for every $k \in [s]$, T_k is a $\times$-rooted canonical ROF, $|\mathrm{var}(T_k)| \leq n$ and $\gamma \in \mathbb{F}$. If for every $k \in [s]$, T_k computes a polynomial of degree at least 2, then $\det(H_{\mathtt{C}}) \neq 0$.*

The above lemma is proved in Section C.1. For our equivalence test, the mere non-zeroness of the Hessian determinant is not enough; we also need some knowledge about its factors. The following claim, proved in Section C.2, gives us some of these factors.

Claim 3.1 (Factors of $\det(H_{\mathtt{C}})$)**.** *Let $\mathbb{F}$ be an arbitrary field and $\mathtt{C}$ a canonical ROF over $\mathbb{F}$. Let Q be a $+$-rooted sub-ROF of $\mathtt{C}$ or a variable connected to a $\times$ gate in $\mathtt{C}$. Let $Q_1, \ldots, Q_m$ be the siblings of Q, i.e., for every $l \in [m]$, Q_l is either a variable or a $+$-rooted sub-ROF of $\mathtt{C}$ and has the same parent as Q. If $|\mathrm{var}(Q_1)| + \cdots + |\mathrm{var}(Q_m)| = e$, then the multiplicity of Q as a factor of $\det(H_{\mathtt{C}})$ is at least $(e-1)$.*

The following corollary is an immediate consequence of the above claim.

Corollary 3.1. *Let $\mathbb{F}$ be an arbitrary field and $\mathtt{C} = T_1 + \cdots + T_s + \gamma$ a canonical ROF over $\mathbb{F}$, where for every $k \in [s]$, T_k is a $\times$-rooted canonical ROF and $\gamma \in \mathbb{F}$. If $k \in [s]$ is such that T_k computes a polynomial of degree at least 3, then there is a $+$-rooted or a variable child Q of T_k such that Q is a factor of $\det(H_{\mathtt{C}})$.*

In the remainder of this section, we describe the variables that are essential for $\det(H_{\mathtt{C}})$ and the variables that do not appear in it. We first define the notion of "skewed paths" which helps us in characterizing these variables.

Definition 3.1 (Skewed path)**.** Let Q be a $+$-rooted sub-ROF of $\mathtt{C}$ and $T_1, \ldots, T_m$ be the product gates on the path from the root of $\mathtt{C}$ to Q. If for all $i \in [m]$, T_i has just two children – a $+$-rooted ROF containing Q and a variable x_i – then we say that the path to Q is *skewed* and identify this path with the "marker" monomial $\mu = \prod_{i\in[m]} x_i$. We say $x_1, \ldots, x_m$ are in the skewed path.

Few other terminologies. We call a variable x a *dangling variable* if its parent in $\mathtt{C}$ is a $+$ gate. For a $+$-rooted sub-ROF $Q = T_1 + \cdots + T_s + \gamma$, where at most one of $T_1, \ldots, T_s$ is a variable and the rest are $\times$-rooted ROFs, we call the sum of all T_i computing a degree two monomial the *quadratic form* of Q. Also, a variable x is said to be in the quadratic form of Q if it is in $\mathrm{var}(T_i)$ for some T_i computing a degree two monomial. Suppose that the path to a $+$-rooted sub-ROF Q is skewed, and the skewed path to Q is identified by the monomial μ. Then, if x is a dangling variable connected to the top-most $+$ gate in Q, we say that x is the *dangling variable along the skewed path* μ. Similarly, we call the quadratic form of Q the *quadratic form along the skewed path* μ. We now describe the essential variables of $\det(H_{\mathtt{C}})$ using these terminologies.

Claim 3.2 (Essential variables of $\det(H_{\mathtt{C}})$)**.** *Let $n \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n$, and $\mathtt{C} = T_1 + \cdots + T_s + \gamma$ be a canonical ROF computing an n-variate polynomial such that for all $k \in [s]$, $\deg(T_k) \geq 2$ and $\gamma \in \mathbb{F}$. Then, every variable in $\mathrm{var}(\mathtt{C})$ other than the variables in the quadratic form of the top-most $+$ gate of $\mathtt{C}$, the dangling variables along skewed paths and the variables appearing in the quadratic forms along skewed paths is truly essential for $\det(H_{\mathtt{C}})$.*

The above claim is proved in Section C.3. The following two claims proved in Sections C.4 and C.5 describe some (but not all) variables that are not present in $\det(H_{\mathtt{C}})$.

Claim 3.3 (Variables of quadratic forms)**.** *Let Q be a $+$-rooted sub-ROF of $\mathtt{C}$ and $\mathbf{y}$ be the set of all variables in the quadratic form of Q. Then, either all $\mathbf{y}$-variables are present in $\det(H_{\mathtt{C}})$ or all are absent. Further if all $\mathbf{y}$-variables are present in $\det(H_{\mathtt{C}})$, then they are also truly essential for $\det(H_{\mathtt{C}})$.*

Claim 3.4 (Missing dangling variables)**.** *Let $\mathtt{C} = T_1 + \cdots + T_s + \gamma$, where $T_1, \ldots, T_s$ are $\times$-rooted sub-ROFs and $\gamma \in \mathbb{F}$. If for any $k \in [m]$, $T_k = xQ$ for a $+$-rooted sub-ROF Q, and y is a dangling variable connected to the top-most $+$ gate of Q, then y is not present in $\det(H_{\mathtt{C}})$.*

4 Equivalence test for ROFs

In this section, we prove Theorem 1. Suppose that we are given black-box access to an $f \in \mathbb{F}[\mathbf{x}]$ in the orbit of an unknown canonical ROF $\mathtt{C}$. We can assume that $\mathtt{C}$ is $+$-rooted: Suppose the root of $\mathtt{C}$ is a $\times$ gate and $\mathtt{C} = g_1 \cdots g_m$, where for every $i \in [m]$, g_i is either a variable or a +-rooted canonical ROF. We obtain black-box access to the irreducible factors $f_1, \ldots, f_{m'}$ of f using the algorithm in [KT90]. Fact 2.5 implies $m = m'$. We can assume that for every $i \in [m]$, $f_i \in \mathrm{orb}(g_i)$ [23]. Then, we apply the algorithm given in Claim 2.3 on $f_1, \ldots, f_m$ to compute an $A_0 \in \mathrm{GL}(n, \mathbb{F})$ such that $f_1(A_0\mathbf{x}), \ldots, f_m(A_0\mathbf{x})$ are pairwise variable disjoint. For $i \in [m]$, let $\mathbf{x}_i = \mathrm{var}(f_i(A_0\mathbf{x}))$ and $f_i'(\mathbf{x}_i) = f_i(A_0\mathbf{x})$. Suppose, for every $i \in [m]$, we could compute an $A_i \in \mathrm{GL}(|\mathbf{x}_i|, \mathbb{F})$ such that

[23] Here we are using a slightly general definition of orbit; see the remark after Definition 2.4.

$f_i'(A_i\mathbf{x}_i) \in$ PS-orb(g_i). Let $A := \text{diag}(A_1, \ldots, A_m)$, which is block-diagonal. Then, $f(A_0 A\mathbf{x}) \in$ PS-orb$(\mathtt{C})$. Thus, the problem reduces to performing equivalence tests for +-rooted canonical ROFs. Before giving the equivalence test, we first give a high-level description of it.

4.1 An overview of the algorithm

We are given black-box access to an $f \in \mathbb{F}[\mathbf{x}]$ such that there exist a $B \in \text{GL}(n, \mathbb{F})$, a $\mathbf{d} \in \mathbb{F}^n$, and a canonical ROF $\mathtt{C}$ satisfying $f = \mathtt{C}(B\mathbf{x} + \mathbf{d})$. Let $\mathtt{C} = T_1 + \cdots + T_s + \gamma$, where at most one of the *terms* $T_1, \ldots, T_s$ is a variable and the rest are $\times$-rooted ROFs, and $\gamma \in \mathbb{F}$. Also, $f = \widehat{T}_1 + \cdots + \widehat{T}_s + \gamma$, where for all $k \in [s]$, $\widehat{T}_k = T_k(B\mathbf{x} + \mathbf{d})$. The equivalence test can be divided into two phases. In the first phase, we compute an $A_0 \in \text{GL}(n, \mathbb{F})$ such that $\widehat{T}_1(A_0\mathbf{x}), \ldots, \widehat{T}_s(A_0\mathbf{x})$ are variable disjoint. In the second phase, we recursively perform equivalence test on the factors of $\widehat{T}_1(A_0\mathbf{x}), \ldots, \widehat{T}_s(A_0\mathbf{x})$. A pictorial overview of the algorithm is given in Appendix G.

Phase 1: Making terms variable disjoint

We rearrange and divide the terms of $\mathtt{C}$ and of f into four groups: Terms $T_1, \ldots, T_{s_1}$ are called the "good" terms of $\mathtt{C}$ if none of them is a dangling variable, nor a degree 2 monomial, nor does it look like $x \cdot Q$ for some $x \in \mathbf{x}$ and a +-rooted ROF Q. Similarly, $\widehat{T}_1, \ldots, \widehat{T}_{s_1}$ are the good terms of f. Terms $T_{s_1+1}, \ldots, T_{s_2}$ are called the "bad" terms of $\mathtt{C}$ if each of them looks like $x \cdot Q$ for some $x \in \mathbf{x}$ and a +-rooted ROF Q; similarly $\widehat{T}_{s_1+1}, \ldots, \widehat{T}_{s_2}$ are the bad terms of f. Observe that the skewed paths in $\mathtt{C}$ occur only in the bad terms of $\mathtt{C}$. If $\mathtt{C}$ has a top dangling variable, then without loss of generality $T_s = x_n$, and $T_{s_2+1} + \cdots + T_{s'}$ is the top quadratic form where $s' = s - 1$. If $\mathtt{C}$ does not have a top dangling variable, then $T_{s_2+1} + \cdots + T_{s'}$ is the top quadratic form where $s' = s$. If $\mathtt{C}$ has a top dangling variable, then let $\ell := \widehat{T}_s$. This phase can be divided into three steps. In the first step, we make all the good terms variable disjoint. In the second step, we make all the bad and quadratic terms variable disjoint and ensure that $\sum_{k=s_2+1}^{s'} \widehat{T}_k$ maps to $(y_1 + c_1)(y_2 + c_2) + \cdots (y_{l-1} + c_{l-1})(y_l + c_l)$ for some $y_1, \ldots, y_l \in \mathbf{x}$ and $c_1, \ldots, c_l \in \mathbb{F}$. If $\mathtt{C}$ has a top dangling variable, then in the third step, we map ℓ to an affine form in a single variable.

Step 1: Making the good terms variable disjoint. To make the terms variable disjoint, we make extensive use of the Hessian determinant. If $\mathtt{C}$ does not have a top dangling variable, then $h = \det(H_f) \neq 0$ (see Lemma 3.1 and Fact 2.8). Otherwise, we apply a random transformation $R \in \mathbb{F}^{n \times n}$ to f and compute $h = \det(H_1)$; here H_1 is the Hessian of $f(R\mathbf{x})$ with respect to $\{x_1, \ldots, x_{n-1}\}$. In this case, we refer to x_n as u_0. If $\mathtt{C}$ does not have a top dangling variable, then let $R = I_{n \times n}$ and $H_1 = H_f$. Let H_2 be the Hessian of $\sum_{k \in [s']} T_k$.[24] Note that H_1 and H_2 are $n \times n$ matrices if $\mathtt{C}$ has no top dangling variable and $(n-1) \times (n-1)$ matrices otherwise. We show in Claim 4.2 that in both cases, h is a non-zero constant multiple of $\det(H_2)(BR\mathbf{x} + \mathbf{d})$. We then invoke Make-Factors-Variable-Disjoint() (see Claim 2.4) on h to compute an $A_0 \in \text{GL}(n, \mathbb{F})$ that makes the factors of h, i.e., $h_1 = \det(H_{T_1})(BR\mathbf{x} + \mathbf{d}), \ldots, h_{s_2} = \det(H_{T_{s_2}})(BR\mathbf{x} + \mathbf{d})$ variable disjoint.[25]

For all $k \in [s_2]$, let $\text{var}(h_k(A_0\mathbf{x})) = \mathbf{z}_k$; as $h_k(A_0\mathbf{x})$ has no redundant variables, all variables in $\mathbf{z}_k$ are essential for it. Let $\mathbf{z}_k' \subseteq \mathbf{z}_k$ be the set of truly essential variables and $\mathbf{z}_k'' := \mathbf{z}_k \setminus \mathbf{z}_k'$ the set of ordinary essential variables in $\mathbf{z}_k$. Let $\mathbf{z} = \uplus_{k \in [s_2]} \mathbf{z}_k$ and $\mathbf{y} = \mathbf{x} \setminus \mathbf{z}$. From Claim 3.2, all variables

[24] We stress that the Hessian of a polynomial g is with respect to $\text{var}(g)$ (unless mentioned otherwise).

[25] We need not mention $\det\left(H_{T_{s_2+1}}\right), \ldots, \det\left(H_{T_{s'}}\right)$ as these are nonzero constants.

in $\mathtt{C}$ other than the top dangling variable, the variables appearing in the top quadratic form, the dangling variables along skewed paths (see Definition 3.1) and the variables appearing in the quadratic forms along skewed paths are truly essential for $\det(H_2)$. In particular, for all good terms T_k, $|\mathrm{var}(T_k)| = |\mathbf{z}'_k| = |\mathbf{z}_k|$; by applying a permutation on the variables in $\mathtt{C}$ if necessary, we can assume that $\mathrm{var}(T_k) = \mathbf{z}'_k = \mathbf{z}_k$. We then argue (using Observation 2.2) that for all $k \in [s_1]$ and all $z \in \mathbf{z}_k$, $BRA_0 \circ z \in \mathbb{F}[\mathbf{z}_k]$. Hence, the good terms $\widehat{T}_1(RA_0\mathbf{x}), \ldots, \widehat{T}_{s_1}(RA_0\mathbf{x})$ are variable disjoint.

We also use Claim 2.1 to compute an affine transformation[26] $C\mathbf{x} + \mathbf{b}$ that maps all the "good" linear factors of $h(A_0\mathbf{x})$ to constant multiples of distinct variables (while preserving the variable disjointness of $\widehat{T}_1(RA_0\mathbf{x}), \ldots, \widehat{T}_{s_1}(RA_0\mathbf{x})$). A linear factor of $h(A_0\mathbf{x})$ is good, if there exists an $x \in \mathbf{x}$ connected to a $\times$ gate (of $\mathtt{C}$) computing a polynomial of degree at least 3 such that $BRA_0\mathbf{x} + \mathbf{d}$ maps x to a constant multiple of that factor. Finally, we update A_0 to RA_0C and $\mathbf{b}$ to $RA_0\mathbf{b}$.

Step 2: Making the bad and quadratic terms variable disjoint. The only variables in a bad term T_k that need not be truly essential for $\det(H_2)$ are the dangling variables along skewed paths and the variables appearing in the quadratic forms along skewed paths – call these the "bad" variables. We show (using Observation 2.2) that all other variables are already mapped to affine forms in $\mathbf{z}_k$ by $BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d}$. Thus, we only need to handle the linear forms that these bad variables map to. Here skewed paths help us. If $z \in \mathbf{z}'_k$ is a variable in a skewed path in T_k and its sibling in T_k is Q, then by "absorbing" an appropriate constant in $Q(BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d})$, we can assume that $BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d}$ maps z to a variable z'.[27] In fact, by permuting the variables in $\mathtt{C}$ if necessary, we can assume that $z' = z$. Hence, each skewed path in $f(A_0\mathbf{x} + \mathbf{b})$ is a "marker" monomial in $\mathbf{z}$.

Step 2.1 (Processing quadratic forms along skewed paths). At first, we treat $f(A_0\mathbf{x} + \mathbf{b})$ as a polynomial in $\mathbf{y} = \mathbf{x} \setminus \mathbf{z}$ over $\mathbb{F}[\mathbf{z}]$ and obtain black-box access to the homogeneous degree-2 component $\widehat{q}$ in $\mathbf{y}$ of $f(A_0\mathbf{x} + \mathbf{b})$. The coefficients of the $\mathbf{y}$-monomials of $\widehat{q}$ are n-sparse polynomials in $\mathbb{F}[\mathbf{z}]$; the monomials of these coefficients correspond to skewed paths and the constant terms of these coefficients originate from the top quadratic form of $\mathtt{C}$. As $\widehat{q}$ is an n^3-sparse polynomial in $\mathbb{F}[\mathbf{z}, \mathbf{y}]$, we can interpolate it using the sparse polynomial interpolation algorithm in [KS01]. Now, by treating $\widehat{q}$ as a polynomial in $\mathbf{z}$ over $\mathbb{F}[\mathbf{y}]$, we see that the coefficients of the $\mathbf{z}$-monomials of $\widehat{q}$ are related to the "unprocessed" quadratic forms along skewed paths as follows.

Let q_0 be the top quadratic form of $\mathtt{C}$, $q_1, \ldots, q_m$ the quadratic forms along skewed paths whose variables do not appear in $\det(H_2)$ (see Claim 3.3), and $\mu_1, \ldots, \mu_m$ the corresponding skewed paths. If $q_i = y_1y_2 + \cdots + y_{l-1}y_l$, then we show that the coefficient of μ_i in $\widehat{q}$ (if $i = 0$, then the $\mathbb{F}[\mathbf{y}]$-constant term in $\widehat{q}$) is $\widetilde{q}_i := [\ell_{y_1}]_{\mathbf{y}} [\ell_{y_2}]_{\mathbf{y}} + \cdots + [\ell_{y_{l-1}}]_{\mathbf{y}} [\ell_{y_l}]_{\mathbf{y}}$. Here, for any $x \in \mathbf{x}$, $\ell_x = BA_0 \circ x$ and $[\ell_x]_{\mathbf{y}}$ is ℓ_x restricted to the $\mathbf{y}$-variables. So, we can use Claim 2.3 and QFE (see Fact A.3) to map the coefficients of all the $\mathbf{z}$-monomials of $\widehat{q}$ to variable disjoint, canonical quadratic forms (i.e., quadratic forms that look like $y_1y_2 + \cdots + y_{l-1}y_l$). We then argue (in Claim 4.5) that if A'_1 is the matrix obtained by combining the matrices output by QFE on $\widetilde{q}_1, \ldots, \widetilde{q}_m$ and $A_1 = A_0A'_1$, then for $\widehat{q}_i := q_i(B\mathbf{x} + \mathbf{d})$, $\widehat{q}_i(A_1\mathbf{x} + \mathbf{b}) = (y_1 + h_1)(y_2 + h_2) + \cdots + (y_{l-1} + h_{l-1})(y_l + h_l)$ for some (hitherto unknown) affine forms $h_1, \ldots, h_l \in \mathbb{F}[\mathbf{z}]$.[28] We can assume that the $\mathbf{y}$-variables in q_i and

[26] Although Phase 1 computes an affine transformation $A_0\mathbf{x} + \mathbf{b}$, it only outputs A_0. Indeed, the terms of $f(A_0\mathbf{x} + \mathbf{b})$ are variable disjoint if and only if the terms of $f(A_0\mathbf{x})$ are variable disjoint.

[27] Absorbing an appropriate constant into Q (i.e., rescaling Q) essentially means that we are starting with a different (but equally valid) B and $\mathbf{d}$. But this is fine as the algorithm is oblivious to the choice of B and $\mathbf{d}$.

[28] The affine form h_i in Step 2.1 should not be confused with the Hessian determinant h_i in Step 1.

$\widehat{q}_i(A_1\mathbf{x}+\mathbf{b})$ are the same by applying a permutation on the variables of $\mathtt{C}$ if necessary.

Step 2.2 (Handling dangling variables along skewed paths). Call the $\mathbf{y}$-variables not appearing in $\widehat{q}_0(A_1\mathbf{x}+\mathbf{b}),\ldots,\widehat{q}_m(A_1\mathbf{x}+\mathbf{b})$ the $\mathbf{u}$-variables. The $\mathbf{u}$-variables appear only in the linear forms corresponding to the dangling variables along skewed paths (that are not truly essential for $\det(H_2)$) and the top dangling variable. We treat $f(A_1\mathbf{x}+\mathbf{b})$ as a polynomial in $\mathbf{u}$ over $\mathbb{F}[\mathbf{x}\setminus\mathbf{u}]$ and obtain black-box access to the homogeneous degree-1 component $\widehat{\ell}$ in $\mathbf{u}$ of $f(A_1\mathbf{x}+\mathbf{b})$. The coefficients of the $\mathbf{u}$-variables of $\widehat{\ell}$ are n-sparse polynomials in $\mathbb{F}[\mathbf{z}]$; the monomials of these coefficients correspond to skewed paths and the constant terms of these coefficients originate from the top dangling variable. As $\widehat{\ell}$ is an n^2-sparse polynomial in $\mathbb{F}[\mathbf{z},\mathbf{u}]$, we interpolate it using [KS01]. Now, by treating $\widehat{\ell}$ as a polynomial in $\mathbf{z}$ over $\mathbb{F}[\mathbf{u}]$, we see that the coefficients of the $\mathbf{z}$-monomials of $\widehat{\ell}$ are related to the "unprocessed" dangling variables along skewed paths as follows.

Let u_0 be the top dangling variable of $\mathtt{C}$, $x_1,\ldots,x_{m'}$ the dangling variables along skewed paths in $\mathtt{C}$ that are not truly essential for $\det(H_2)$, and $\mu_1,\ldots,\mu_{m'}$ the corresponding skewed paths. Then, $[\ell_{x_i}]_{\mathbf{u}}$ (where ℓ_{x_i} is the linear form that $BA_1\mathbf{x}$ maps x_i to) is the coefficient of μ_i in $\widehat{\ell}$ and the $\mathbb{F}[\mathbf{u}]$-constant term in $\widehat{\ell}$ is $[\ell_{u_0}]_{\mathbf{u}}$. We then find a basis $\mathscr{B}$ of the coefficients of $\mathbf{z}$-monomials (which are linear forms in $\mathbf{u}$) and compute an A_2' that maps these basis elements to distinct $\mathbf{u}$-variables. Now there is a *subtle point* to address here: The size of $\mathscr{B}$ can possibly be strictly less than $m'+1$, which is the number of "unprocessed" dangling variables. And yet we wish to show that A_2' takes care of all the $m'+1$ dangling variables. Let us see (at a high level) how this works.

We argue that the elements of $\mathscr{B}$ are of two kinds. First, we show (in Observation 4.2) that for $x\in\{u_0,x_1,\ldots,x_{m'}\}$, $[\ell_x]_{\mathbf{u}}$ is always in $\mathscr{B}$ if x does not appear in $\det(H_2)$; this part of $\mathscr{B}$ is independent of the choice of the basis $\mathscr{B}$. Second, we show in Claim 4.7 that the remaining elements of $\mathscr{B}$ correspond to a set of redundant variables of $\det(H_2)$ among the variables appearing in $\det(H_2)$; this part varies with the choice of $\mathscr{B}$. This structure of $\mathscr{B}$ helps us prove the following: For $k\in\{s_1+1,\ldots,s_2\}$, let $\mathbf{x}_k'$ be the set of all $x\in\mathrm{var}(T_k)$ such that $[\ell_x]_{\mathbf{u}}$ is in $\mathscr{B}$. Let $\mathbf{y}_k$ be the union of the $\mathbf{y}$-variables present in the quadratic forms along skewed paths in $\widehat{T}_k(A_1A_2'\mathbf{x}+\mathbf{b})$ and the $\mathbf{u}$-variables in $\ell_x(A_2'\mathbf{x})$ for $x\in\mathbf{x}_k'$. Then, we show in Observation 4.4 that as long as we map $\ell_x(A_2'\mathbf{x})$ to a linear form in $\mathbb{F}[\mathbf{z}_k\uplus\mathbf{y}_k]$ for all $x\in\mathbf{x}_k'$, we would have mapped all $\ell_{x'}$, where $x'\in\mathrm{var}(T_k)$ is a dangling variable along a skewed path, to linear forms in $\mathbb{F}[\mathbf{z}_k\uplus\mathbf{y}_k]$.

It follows that $[\ell_{u_0}]_{\mathbf{u}}$ is in $\mathscr{B}$ (as $u_0\notin\mathrm{var}(\det(H_2))$) and A_2' maps it to u_0 (without loss of generality by applying a permutation on $\mathrm{var}(\mathtt{C})$ if required). Apart from $[\ell_{u_0}]_{\mathbf{u}}$, let $[\ell_{x_1}]_{\mathbf{u}},\ldots,[\ell_{x_m}]_{\mathbf{u}}$ be the other $\mathscr{B}$-elements. Then, $\ell_{u_0}(A_2'\mathbf{x})$ looks like $u_0+h_{0,1}(\mathbf{y}\setminus\mathbf{u})+h_{0,2}(\mathbf{z})$ and $\ell_{x_i}(A_2'\mathbf{x})$ looks like $u_i+h_{i,1}(\mathbf{y}\setminus\mathbf{u})+h_{i,2}(\mathbf{z})$, where $h_{i,1}(\mathbf{y}\setminus\mathbf{u})$ is an affine form in $\mathbf{y}\setminus\mathbf{u}$ and $h_{i,2}(\mathbf{z})$ is an affine form in $\mathbf{z}$ for all $0\le i\le m$. In fact, by renaming the variables of $\mathtt{C}$ if required, we can assume $x_i=u_i$ for all $i\in[m]$. So, we will refer to $\mathbf{x}_k'$ as $\mathbf{u}_k$. We save $V:=\{(1,u_0),(\mu_1,u_1),\ldots,(\mu_m,u_m)\}$ for Step 2.3. Let $A_2=A_1A_2'$ and ℓ_x be the linear form that BA_2 maps x to. Our goal in the next step is to compute a linear transformation that for all $k\in\{s_1+1,\ldots,s_2\}$ removes "external" variables, i.e., variables not in $\mathbf{z}_k\uplus\mathbf{y}_k$, from all ℓ_y and ℓ_u for $y\in\mathbf{y}_k\setminus\mathbf{u}_k$ and $u\in\mathbf{u}_k$. Also, we want to map the top quadratic form to an expression $(y_1+c_1)(y_2+c_2)+\cdots(y_{l-1}+c_{l-1})(y_l+c_l)$, where $c_i\in\mathbb{F}$.

Step 2.3 (Removing external variables from terms). We first remove external $\mathbf{z}$-variables from ℓ_y for $y\in\mathbf{y}\setminus\mathbf{u}$; such an ℓ_y does not have external $\mathbf{y}$-variables. We also remove external $(\mathbf{y}\setminus\mathbf{u})$-variables from ℓ_u for $u\in\mathbf{u}$; such an ℓ_u does not have external $\mathbf{u}$-variables. This is done as follows: For all

$y \in \mathbf{y} \setminus \mathbf{u}$, compute $g = \frac{\partial f(A_2\mathbf{x}+\mathbf{b})}{\partial y}$; it will contain only one $\mathbf{y}$-variable, say y'. The sparsity of g is at most $2n$, so we can interpolate it. If y' is multiplied by the $\mathbf{z}$-monomial μ in g (μ can be 1), then we express g as $\mu(y' + \ell') + r(\mathbf{z})$, where ℓ' is an affine form in $\mathbf{z}$, and $r(\mathbf{z}) \in \mathbb{F}[\mathbf{z}]$. Suppose $y \in \mathbf{y}_k \setminus \mathbf{u}_k$ (k can be figured out from μ). We show in Observation 4.5 that for every $z \notin \mathbf{z}_k$ in ℓ', the coefficient β of z in ℓ' can be assumed to be the same as its coefficient in $\ell_{y'}$. So, by translating y' by $-\beta z$ we can remove z from $\ell_{y'}$. Then, we show in Observation 4.6 that for all monomials μ_i in $r(\mathbf{z})$ (μ_i can be 1) such that $(\mu_i, u_i) \in V$ and $u_i \notin \mathbf{u}_k$, the coefficient β of μ_i in $r(\mathbf{z})$ can be assumed to be the same as the coefficient of y in ℓ_{u_i}. So, to remove y from the latter, we just need to translate u_i by $-\beta y$. The transformation A_3' computed thus removes external $\mathbf{z}$-variables from ℓ_y for $y \in \mathbf{y} \setminus \mathbf{u}$ and external $(\mathbf{y} \setminus \mathbf{u})$-variables from ℓ_u for $u \in \mathbf{u}$. It also follows from the disambiguation argument in Observations 4.5 and 4.6 that A_3' maps the top quadratic form to $(y_1 + c_1)(y_2 + c_2) + \cdots (y_{l-1} + c_{l-1})(y_l + c_l)$ for $c_i \in \mathbb{F}$.

Let $A_3 = A_2A_3'$ and ℓ_x be the linear form that BA_3 maps x to. Then, we only need to remove the external $\mathbf{z}$-variables from ℓ_u for all $u \in \mathbf{u}_k$ and $k \in \{s_1 + 1, \ldots, s_2\}$. Let $u_i \in \mathbf{u}_k$. To remove $z \notin \mathbf{z}_k$ from ℓ_{u_i}, we obtain g from $f(A_3\mathbf{x} + \mathbf{b})$ by setting all variables other than $\mathrm{var}(\mu_i)$ and z to 0. Then, using the disambiguation argument in Observation 4.7, we show that the coefficient β of z in ℓ_{u_i} can be readily derived from the coefficient of μ_i in $\frac{\partial g}{\partial z}$. Thus, by translating u_i by $-\beta z$, we can remove z from the ℓ_{u_i}. If A_4' is the transformation computed this way and $A_4 = A_3A_4'$, then in $f(A_4\mathbf{x} + \mathbf{b})$ all non-linear terms are variable disjoint.

Step 3: Learning the top linear form. Let ℓ_x be the linear form that BA_4 maps x to. If $\mathtt{C}$ has a top dangling variable, then Steps 1 and 2 ensure that u_0 is only present in the linear form ℓ_{u_0}. Moreover, Step 2.3 implies that ℓ_{u_0} is free of $\mathbf{y} \setminus \{u_0\}$ variables. In particular, none of the variables in the quadratic term $\sum_{k=s_2+1}^{s'} \widehat{T}_k(A_4\mathbf{x} + \mathbf{b})$ is in ℓ_{u_0}. So, we only need to remove the variables in $\widehat{T}_1(A_4\mathbf{x} + \mathbf{b}), \ldots, \widehat{T}_{s_2}(A_4\mathbf{x} + \mathbf{b})$ from ℓ_{u_0}. Towards this, we first use second derivatives (in Claim 4.13) to learn $\mathrm{var}(\widehat{T}_1(A_4\mathbf{x} + \mathbf{b})), \ldots, \mathrm{var}(\widehat{T}_{s_2}(A_4\mathbf{x} + \mathbf{b}))$; let these variable sets be $\mathbf{z}_1, \ldots, \mathbf{z}_{s_2}$.[29] Then, we iteratively learn ℓ_{u_0} in s_2 iterations. In the k-th iteration, we learn $[\ell_{u_0}]_{\mathbf{z}_k}$. To do this, we first obtain $\widehat{T} := \widehat{T}_k(A_4\mathbf{x} + \mathbf{b}) + [\ell_{u_0}]_{\mathbf{z}_k} + \gamma'$, where $\gamma' \in \mathbb{F}$, by setting all but $\mathbf{z}_k$-variables to zero. The argument to learn $[\ell_{u_0}]_{\mathbf{z}_k}$ from $\widehat{T}$ is a generalization of the argument used for '*Finding c*' under Hurdle 3 in Section 1.3.2. However, it is more involved as (unlike the constant c in Section 1.3.2) $[\ell_{u_0}]_{\mathbf{z}_k}$ is *not* uniquely determined. This leads to a couple of complications: One, the circuit that we derive by "learning" $[\ell_{u_0}]_{\mathbf{z}_k}$ is strictly speaking not canonical. Two, it is now unclear how to test if a chosen factor of the Hessian determinant of $\widehat{T}$ is "good". We elaborate on these next to show how the non-uniqueness of $[\ell_{u_0}]_{\mathbf{z}_k}$ is handled.

We use the factors of h', the Hessian determinant of $\widehat{T}$ with respect to the $\mathbf{z}_k$-variables, to learn an affine form ℓ_k such that $\widehat{T} - \ell_k$ is reducible. Observe that h' is the Hessian determinant of T_k evaluated at $BA_4\mathbf{x} + B\mathbf{b} + \mathbf{d}$. Corollary 3.1 implies that at least one of the factors of $\widehat{T}_k(A_4\mathbf{x} + \mathbf{b})$ is a factor of h'. We will refer to the factors of h' that are also factors of $\widehat{T}_k(A_4\mathbf{x} + \mathbf{b})$ as "good" factors of h'. Let $\widehat{Q}$ be a constant multiple of a good factor of h'. We now show how to learn ℓ_k using $\widehat{Q}$.

$\widehat{Q}$ is not linear. Let $\mathbf{a}_1, \ldots, \mathbf{a}_{|\mathbf{z}_k|}$ be random $\mathbb{F}$-vectors of size $|\mathbf{z}_k|$ and t be a fresh variable. For all

[29] Here, we are overloading the notation a bit. In Steps 1 and 2, $\mathbf{z}_k$ was the variables of the Hessian determinant of T_k evaluated at $BRA_0\mathbf{x} + \mathbf{d}$. In other words, the new set of $\mathbf{z}_k$ variables is the disjoint union of the old $\mathbf{z}_k$ and $\mathbf{y}_k$.

$i \in [|\mathbf{z}_k|]$, interpolate $\widehat{Q}(t\mathbf{a}_i)$ and $\widehat{T}(t\mathbf{a}_i)$. Discover $\widehat{Q}'_i(t)$ of degree at most n and $\beta_{i,0},\ \beta_{i,1} \in \mathbb{F}$ such that $\widehat{Q}(t\mathbf{a}_i) \cdot \widehat{Q}'_i(t) + \beta_{i,1} \cdot t + \beta_{i,0} = \widehat{T}(t\mathbf{a}_i)$ by solving a system of linear equations in the coefficients of $\widehat{Q}'_i(t)$ and $\beta_{i,0},\ \beta_{i,1}$. One solution is $\widehat{Q}'_i = \left(\widehat{T}_k(A_4\mathbf{x}+\mathbf{b})/\widehat{Q}\right)(t\mathbf{a}_i)$, $\beta_{i,1} = [\ell_{u_0}]_{\mathbf{z}_k}(\mathbf{a}_i)$, and $\beta_{i,0} = \gamma'$. We show in the proof of Claim 4.15 that this solution is unique with high probability over the randomness of $\mathbf{a}_1, \ldots, \mathbf{a}_{|\mathbf{z}_k|}$. Then, we set ℓ_k to be the affine form obtained by interpolation using $\beta_{1,1}, \ldots, \beta_{|\mathbf{z}_k|,1}$ and $\beta_{i,0} = \gamma'$. Hence, $\ell_k = [\ell_{u_0}]_{\mathbf{z}_k} + \gamma'$, and $\widehat{T} - \ell_k = \widehat{T}_k(A_4\mathbf{x}+\mathbf{b})$ is reducible.

$\widehat{Q}$ is linear. Suppose that $\widehat{Q} = z$ (recall that in Step 1, we would have mapped $\widehat{Q}$ to a single variable). Let $\mathbf{a}_1, \ldots, \mathbf{a}_{|\mathbf{z}_k|-1}$ be random $\mathbb{F}$-vectors of size $|\mathbf{z}_k| - 1$ and t a fresh variable. For all $i \in [|\mathbf{z}_k| - 1]$, interpolate the bivariate polynomial $\widehat{T}(z, \mathbf{z}_k \setminus \{z\} = t\mathbf{a}_i)$. Find $\widehat{Q}'_i(z,t)$ of degree at most n and $\beta_{i,0},\ \beta_{i,1},\ \beta_{i,2} \in \mathbb{F}$ such that $z\widehat{Q}'_i(z,t) + \beta_{i,2} \cdot z + \beta_{i,1} \cdot t + \beta_{i,0} = \widehat{T}(z, \mathbf{z}_k \setminus \{z\} = t\mathbf{a}_i)$ by solving a system of linear equations in the coefficients of $\widehat{Q}'_i(z,t)$ and $\beta_{i,0},\ \beta_{i,1},\ \beta_{i,2}$. One such solution is $\widehat{Q}'_i = \left(\widehat{T}_k(A_4\mathbf{x}+\mathbf{b})/\widehat{Q}\right)(z, t\mathbf{a}_i)$, $\beta_{i,2} = c_z$, the coefficient of z in $[\ell_{u_0}]_{\mathbf{z}_k}$, $\beta_{i,1} = [\ell_{u_0}]_{\mathbf{z}_k \setminus \{z\}}(\mathbf{a}_i)$ and $\beta_{i,0} = \gamma'$. We show in the proof of Claim 4.16 that $\beta_{i,1}$ and $\beta_{i,0}$ are unique with high probability over the randomness of $\mathbf{a}_1, \ldots, \mathbf{a}_{|\mathbf{z}_k|-1}$. So, after using $\beta_{1,1}, \ldots, \beta_{|\mathbf{z}_k|-1,1}$ and $\beta_{i,0} = \gamma'$ to interpolate an affine form ℓ_k, we get $\ell_k = [\ell_{u_0}]_{\mathbf{z}_k \setminus \{z\}} + \gamma'$. Hence, $\widehat{T} - \ell_k = z(\widehat{T}_k(A_4\mathbf{x}+\mathbf{b})/z + c_z)$ is reducible.

What if $\widehat{Q}$ is not good? As at least one factor $\widehat{Q}$ of h' is good, by iterating over all its factors, we ultimately find an ℓ_k such that $\widehat{T} - \ell_k$ is reducible. If $\widehat{Q}$ is good, then either $\ell_k = [\ell_{u_0}]_{\mathbf{z}_k} + \gamma'$ and $\widehat{T} - \ell_k = \widehat{T}_k(A_4\mathbf{x}+\mathbf{b})$ or $\ell_k = [\ell_{u_0}]_{\mathbf{z}_k \setminus \{z\}} + \gamma'$ and $\widehat{T} - \ell_k = z(\widehat{T}_k(A_4\mathbf{x}+\mathbf{b})/z + c_z)$ (where $\widehat{Q} = z$). But, what if $\widehat{Q}$ is not good? It turns out that the algorithm only needs to care about finding an ℓ_k such that $\widehat{T} - \ell_k$ is reducible; such an ℓ_k is always the desired one. This is because, it is implied by the proof of Claim 4.17 that if $\widehat{T} - \ell_k$ is reducible, then the following holds: If $\widehat{T}_k(A_4\mathbf{x}+\mathbf{b})$ has no linear factors, then $\ell_k = [\ell_{u_0}]_{\mathbf{z}_k} + \gamma'$. On the other hand, if $\widehat{T}_k(A_4\mathbf{x}+\mathbf{b}) = \widehat{Q}_{k,1} \cdots \widehat{Q}_{k,m_k}$ and $\widehat{Q}_{k,1} = z$, then ℓ_k and $[\ell_{u_0}]_{\mathbf{z}_k}$ must agree on the coefficients of all variables in $\mathbf{z}_k$ except perhaps that of z. In both the cases, $\widehat{T} - \ell_k = \widehat{Q}_{k,1}\left(\widehat{Q}_{k,2} \cdots \widehat{Q}_{k,m_k} + c\right)$ for some $c \in \mathbb{F}$. Thus, if we redefine $\widehat{T}_k(A_4\mathbf{x}+\mathbf{b})$ as $\widehat{Q}_{k,1}\left(\widehat{Q}_{k,2} \cdots \widehat{Q}_{k,m_k} + c\right)$, then $\widehat{T}_k(A_4\mathbf{x}+\mathbf{b}) = T'_k(BA_4\mathbf{x} + B\mathbf{b} + \mathbf{d})$, where if $T_k = Q_{k,1} \cdots Q_{k,m_k}$ then $T'_k := Q_{k,1}(Q_{k,2} \cdots Q_{k,m_k} + c)$. Let $\mathtt{C}'$ be obtained from $\mathtt{C}$ by replacing T_k with T'_k whenever necessary. If A_5 is obtained from A_4 by translating u_0 by the linear part of $\sum_{k \in s_2} \ell_k$, then $f(A_5\mathbf{x}+\mathbf{b}) = \widehat{T}_1(A_5\mathbf{x}+\mathbf{b}) + \cdots + \widehat{T}_{s'}(A_5\mathbf{x}+\mathbf{b}) + u_0 + \gamma$ for $\gamma \in \mathbb{F}$, and $f(A_5\mathbf{x}+\mathbf{b}) = \mathtt{C}'(BA_5\mathbf{x} + B\mathbf{b} + \mathbf{d})$. Notice that all the terms of $f(A_5\mathbf{x}+\mathbf{b})$ are variable disjoint.

Re-canonizing the ROF. Circuit $\mathtt{C}'$ need not be a canonical ROF as Property 6 of Definition 2.6 may fail. However, for all $k \in [s_2]$, all the factors of T'_k are canonical. As we recursively perform ET on only the factors of $\widehat{T}_k(A_5\mathbf{x})$, $\mathtt{C}'$ not being canonical is not a problem during recursion. However, at the end of the recursion, we are left with reconstructing a canonical ROF where Property 6 may not hold. But this is not an issue as the ROF reconstruction algorithm (Algorithm 13) in Section F works for canonical ROFs that may not satisfy Property 6. Once an ROF is constructed using Algorithm 13, we go over the ROF in linear time to ensure that Property 6 is satisfied.

Phase 2: Recursively performing equivalence test on the factors of variable disjoint terms

To perform equivalence test on the factors $\widehat{Q}_{k,1}(A_5\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(A_5\mathbf{x})$ of $\widehat{T}_k(A_5\mathbf{x})$ we need to obtain black-box access to each of the factors using *only one query* to the black-box of f. It is important that a single query to f is used, or else, the running time of the algorithm will blow up exponentially with the product depth of the ROF. A detailed overview of this phase is already provided in the discussion following Hurdle 3 in Section 1.3.2. Other details are given in Sections 4.3.4 and 4.3.6.

4.2 The algorithm

Having given a high level overview of the algorithm, we now describe it formally.

Algorithm 1 Find-Equivalence($f(\mathbf{x})$)

Input: Black-box access to an n-variate polynomial f in the orbit of an unknown $+$-rooted canonical ROF $\mathtt{C}$ such that every $x \in \mathbf{x}$ is essential for f.
Output: An $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f(A\mathbf{x}) \in \text{PS-orb}(\mathtt{C})$.

/* The base case. */

1. If $\deg(f) = 1$, return $I_{n\times n}$.

/* Making the non-linear terms of f variable disjoint. */

2. $A_0, \mathbf{b}, \mathbf{z}, \{u_0\}, \{\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_m\} \leftarrow$ Make-Good-Terms-Var-Disjoint(f) (Procedure 2).
3. $A \leftarrow$ Make-Bad-And-Quadratic-Terms-Var-Disjoint($f(\mathbf{x}), A_0, \mathbf{b}, \mathbf{z}, u_0$) (Procedure 3) .

/* Learning var $\left(\widehat{T}_1(A\mathbf{x}+\mathbf{b})\right), \ldots,$ var $\left(\widehat{T}_{s_2}(A\mathbf{x}+\mathbf{b})\right)$. */

4. $\mathbf{y} \leftarrow \mathbf{x} \setminus \mathbf{z}$. $E \leftarrow \emptyset$, $G \leftarrow (\{\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_m\} \uplus \mathbf{y}, E)$, a graph.
5. **for** $i,\ j \in [m]$ **do**
6. If for any $z_1 \in \widehat{\mathbf{z}}_i$ and $z_2 \in \widehat{\mathbf{z}}_j$, $\frac{\partial^2 f(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial z_2} \neq 0$, add edge $\{\widehat{\mathbf{z}}_i, \widehat{\mathbf{z}}_j\}$ to E.
7. **end for**
8. **for** $i \in [m]$ and $y \in \mathbf{y}$ **do**
9. If for any $z \in \widehat{\mathbf{z}}_i$ and $y \in \mathbf{y}$, $\frac{\partial^2 f(A\mathbf{x}+\mathbf{b})}{\partial z \partial y} \neq 0$, add edge $\{\widehat{\mathbf{z}}_i, y\}$ to E.
10. **end for**
11. $\mathbf{z}_1, \ldots, \mathbf{z}_{s_2} \leftarrow$ the variable sets of size more than 1 corresponding to the different connected components of G. $\mathbf{z} \leftarrow \uplus_{k=1}^{s_2} \mathbf{z}_k$, $\mathbf{y} \leftarrow \mathbf{x} \setminus \mathbf{z}$.

/* Learning the top linear form if it exits. */

12. **if** $\{u_0\} \neq \emptyset$ **then**
13. $\ell' \leftarrow$ Find-Top-Linear-Form($f(A\mathbf{x}+\mathbf{b})$) (Procedure 5). Update A to map u_0 to $u_0 - \ell'$.
14. **end if**

15. **for** $k \in [s_2]$ **do**
16. $\widehat{T} \leftarrow$ Compute-Term-Black-Box($f\left(A\left(\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k = \mathbf{0}\right)\right)$) (Procedure 6).

/* Making the factors of $\widehat{T}_k(A\mathbf{x})$ variable disjoint. */

17. $\widehat{Q}_1, \ldots, \widehat{Q}_{m_k} \leftarrow$ black-boxes of the factors of $\widehat{T}$ obtained using the algorithm in [KT90].

18. Compute an $A_{k,0} \in \mathrm{GL}(|\mathbf{z}_k|, \mathbb{F})$ s.t. $\widehat{Q}_1(A_{k,0}\mathbf{z}_k), \ldots, \widehat{Q}_{m_k}(A_{k,0}\mathbf{z}_k)$ are variable disjoint using Make-Polys-Var-Disjoint() (see Claim 2.3). $\forall l \in [m_k]$, $\widehat{Q}_l \leftarrow \widehat{Q}_l\left(A_{k,0}\mathbf{z}_k\right)$, $\mathbf{z}_{k,l} \leftarrow \mathrm{var}\left(\widehat{Q}_l\right)$.

/* Performing equivalence test on $\widehat{Q}_1, \ldots, \widehat{Q}_{m_k}$. */

19. $F \leftarrow$ a subset of $\mathbb{F}$ of size n^5.[30] $\mathbf{a} \leftarrow$ a vector of size $\mathbf{z}_k$ containing random elements from F.
20. **for** $l \in [m_k]$ **do**
21. $\quad \mathbf{a}' \leftarrow \mathbf{a}$ restricted to entries corresponding to $\mathbf{z} \setminus \mathbf{z}_{k,l}$.
22. $\quad \beta_l \leftarrow \prod_{l' \in [m_k] \setminus \{l\}} \widehat{Q}_{l'}\left(\mathbf{z}_{k,l}, \mathbf{z}_k \setminus \mathbf{z}_{k,l} = \mathbf{a}'\right)$. $\widehat{Q}_{l'} \leftarrow \beta_l^{-1} \cdot \widehat{T}\left(A_{k,0}\left(\mathbf{z}_{k,l}, \mathbf{z}_k \setminus \mathbf{z}_{k,l} = \mathbf{a}'\right)\right)$. $A_{k,l} \leftarrow$ Find-Equivalence$\left(\widehat{Q}_l\right)$.
23. **end for**
24. Construct an $A'_{k,0} \in \mathrm{GL}(|\mathbf{z}_k|, \mathbb{F})$ that maps every $z \in \mathbf{z}_{k,l}$ to $A_{k,l} \circ z, \forall l \in [m_k]$. $A_k \leftarrow A_{k,0}A'_{k,0}$.
25. **end for**
26. Construct an $A' \in \mathrm{GL}(n, \mathbb{F})$ that maps all $z \in \mathbf{z}_k$ to $A_k \circ z$, $\forall k \in [s_2]$ and all $y \in \mathbf{y}$ to y. $A \leftarrow AA'$.
27. **if** $\{u_0\} \neq \emptyset$ **then**
28. $\quad f' \leftarrow$ Reconstruct-ROF($f(A\mathbf{x})$) (see Appendix F, Algorithm 13).
29. $\quad$ For every term of f' that looks like $(\alpha_1 x + \alpha_0)Q$ and Q has a constant β attached to the top $+$ gate but not a dangling variable, modify A to map u_0 to $u_0 - \alpha_1 \cdot \beta \cdot x$.
30. **end if**
31. Return A.

Recall that the algorithm is given black-box access to an $f \in \mathbb{F}[\mathbf{x}]$ such that there exist a $B \in \mathrm{GL}(n, \mathbb{F})$, a $\mathbf{d} \in \mathbb{F}^n$, and a canonical ROF $\mathtt{C}$ satisfying $f = \mathtt{C}(B\mathbf{x} + \mathbf{d})$. Also, there are no redundant variables in f. Further $\mathtt{C} = T_1 + \cdots + T_s + \gamma$, where $T_1, \cdots, T_s$ are $\times$-rooted canonical ROFs and $\gamma \in \mathbb{F}$. Also, $f = \widehat{T}_1 + \cdots + \widehat{T}_s + \gamma$, where for all $k \in [s]$, $T_k(B\mathbf{x} + \mathbf{d}) = \widehat{T}_k$. $T_1, \ldots, T_{s_1}$ are the "good" terms of $\mathtt{C}$, i.e. none of them is a dangling variable, nor a degree 2 monomial, nor does it look like $x \cdot Q$ for some $x \in \mathbf{x}$ and a $+$-rooted ROF Q. Similarly, $\widehat{T}_1, \ldots, \widehat{T}_{s_1}$ are the good terms of f. $T_{s_1+1}, \ldots, T_{s_2}$ are the "bad" terms of $\mathtt{C}$, i.e. they look like $x \cdot Q$, while $\widehat{T}_{s_1+1}, \ldots, \widehat{T}_{s_2}$ are the bad terms of f. If $\mathtt{C}$ has a top dangling variable, $T_s = x_n$, $s' := s - 1$, and $T_{s_2+1} + \cdots + T_{s-1}$ is the top quadratic form. If $\mathtt{C}$ does not have a top dangling variable, then $T_{s_2+1} + \cdots + T_s$ is the top quadratic form and $s' := s$. If $\mathtt{C}$ has a top dangling variable, then $\ell := \widehat{T}_s$.

We shall give a formal description of the procedures Make-Good-Terms-Var-Disjoint(), Make-Bad-And-Quadratic-Terms-Var-Disjoint(), Find-Top-Linear-Form(), and Compute-Term-Black-Box() while analysing the algorithm. Make-Good-Terms-Var-Disjoint() outputs an $A_0 \in \mathrm{GL}(n, \mathbb{F})$ and a $\mathbf{b} \in \mathbb{F}^n$ such that $\widehat{T}_1(A_0\mathbf{x}), \ldots, \widehat{T}_{s_1}(A_0\mathbf{x})$ are variable disjoint. Make-Bad-And-Quadratic-Terms-Var-Disjoint() outputs an $A \in \mathrm{GL}(n, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^n$ such that $\widehat{T}_1(A\mathbf{x} + \mathbf{b}), \ldots, \widehat{T}_{s'}(A\mathbf{x} + \mathbf{b})$ are variable disjoint and $\sum_{k=s_2+1}^{s'} \widehat{T}_k(A\mathbf{x} + \mathbf{b}) = \sum_{k=s_2+1}^{s'} (y_{k,1} + c_{k,1})(y_{k,2} + c_{k,2})$, where $c_{k,1}, c_{k,2} \in \mathbb{F}$. Find-Top-Linear-Form() maps $\ell(A\mathbf{x})$ to a single variable u_0. Compute-Term-Black-Box() helps obtain black-box access to $\widehat{T}_1(A\mathbf{x}), \ldots, \widehat{T}_{s_2}(A\mathbf{x})$ using just one query to the black-box of f. Finally Steps 15-25 recursively perform equivalence test on the factors of $\widehat{T}_1(A\mathbf{x}), \ldots, \widehat{T}_{s_2}(A\mathbf{x})$.

[30] Here n^5 is somewhat arbitrary. We simply want to ensure that after we apply union bound to the error probabilities in different steps of the algorithm, the total error probability is still small.

4.3 Analysis of the algorithm

In this section, we prove the following lemma. This lemma along with the analysis of the running time in Section 4.3.6 proves Theorem 1.

Lemma 4.1 (Correctness of Algorithm 1)**.** *Let $\mathbb{F}$ be a field of* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\geq n^2$ *and* $|\mathbb{F}| \geq n^{13}$*. Given black-box access to an n-variate polynomial f in the orbit of an unknown $+$-rooted canonical ROF $\mathtt{C}$ such that every $x \in \mathbf{x}$ is essential for f, Algorithm 1 outputs an $A \in \mathrm{GL}(n, \mathbb{F})$ such that $f(A\mathbf{x}) \in \text{PS-orb}(\mathtt{C})$.*

Towards proving this lemma, we first formally describe the Make-Good-Terms-Var-Disjoint(), Make-Bad-And-Quadratic-Terms-Var-Disjoint(), Find-Top-Linear-Form(), and Compute-Term-Black-Box() procedures in the following sections.

4.3.1 Making the good terms variable disjoint

The following procedure is used to make all the good terms variable disjoint.

Procedure 2 Make-Good-Terms-Var-Disjoint($f(\mathbf{x})$)

Input. $f(\mathbf{x})$, a polynomial in the orbit of a $+$-rooted canonical ROF $\mathtt{C}$.
Output.

1. $A_0 \in \mathrm{GL}(n, \mathbb{F}), \mathbf{b} \in \mathbb{F}^n$ such that for $k \neq k' \in [s_1]$, $\mathrm{var}(\widehat{T}_k(A_0\mathbf{x}+\mathbf{b})) \cap \mathrm{var}(\widehat{T}_{k'}(A_0\mathbf{x}+\mathbf{b})) = \emptyset$.
2. For all $x \in \mathbf{x}$ connected to a $\times$-gate in $\mathtt{C}$ computing a polynomial of degree ≥ 3, $BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d}$ maps x to constant multiple of a variable. If $\mathtt{C}$ has a top dangling variable, then $u_0 = x_n$.
3. $\{\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_m\}$ such that there exists a partition $I_1, \ldots, I_{s_2}$ of $[m]$ such that for all $k \in [s_2]$, $\uplus_{i \in I_k} \widehat{\mathbf{z}}_i = \mathrm{var}\left(\det(H_{T_k})(BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d})\right)$. $\mathbf{z} = \uplus_{i \in [m]} \widehat{\mathbf{z}}_i$.

/* Computing the Hessian determinant of f. */

1. **if** $\det(H_f) = 0$ **then**
2. $F \leftarrow$ a subset of $\mathbb{F}$ of size at least n^5. $R \leftarrow$ an $n \times n$ random matrix with entries picked independently and uniformly from F. $u_0 \leftarrow x_n$.
3. $h \leftarrow$ the Hessian determinant of $f(R\mathbf{x})$ with respect to $\mathbf{x} \setminus \{u_0\}$ variables.
4. $A_0 \leftarrow$ Remove-Redundant-Vars(h, u_0) (see Claim 2.2).
5. **else**
6. $\{u_0\} \leftarrow \emptyset$, $h \leftarrow \det(H_f)$, $R \leftarrow I_{n\times n}$, $A_0 \leftarrow I_{n\times n}$.
7. **end if**
8. $A_0', \widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_m \leftarrow$ Make-Factors-Var-Disjoint$(h(A_0\mathbf{x}))$ (see Claim 2.4). $A_0 \leftarrow A_0A_0'$. $\mathbf{z} \leftarrow \mathrm{var}\left(h(A_0\mathbf{x})\right)$.

/* Mapping the good linear factors of $h(A_0\mathbf{x})$ to distinct variables. */

9. $V \leftarrow$ the set of all linear factors of $h(A_0\mathbf{x})$. $C \leftarrow I_{n\times n}$, $\mathbf{b}' \leftarrow \mathbf{0}$.
10. **for** $\ell' \in V$ **do**
11. If $N_{ess}\left(f(RA_0\mathbf{x}) \mod \ell'\right) < n-2$, pick a $z \in \mathrm{var}(\ell')$ and update C and $\mathbf{b}'$ such that $\ell(C\mathbf{x} + \mathbf{b}') = z$ (see Claim 2.1).
12. **end for**
13. $\mathbf{b} \leftarrow RA_0\mathbf{b}'$, $A_0 \leftarrow RA_0C$. Return $A_0, \mathbf{b}, \mathbf{z}, \{u_0\}, \{\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_m\}$.

We now prove the following lemma which establishes the correctness of the above procedure.

Lemma 4.2 (Correctness of Procedure 2). *Make-Good-Terms-Variable-Disjoint($f(\mathbf{x})$) outputs an $A_0 \in \mathrm{GL}(n, \mathbb{F})$ and a $\mathbf{b} \in \mathbb{F}^n$ such that $\widehat{T}_1(A_0\mathbf{x}+\mathbf{b}), \ldots, \widehat{T}_{s_1}(A_0\mathbf{x}+\mathbf{b})$ are pairwise variable disjoint. Further for all $x \in \mathbf{x}$ connected to a $\times$-gate in $\mathtt{C}$ computing a polynomial of degree at least 3, $BA_0\mathbf{x}+B\mathbf{b}+\mathbf{d}$ maps x to constant multiple of a variable. Moreover, there exists a partition $I_1, \ldots, I_{s_2}$ of $[m]$ such that for all $k \in [s_2]$, $\uplus_{i\in I_k}\widehat{\mathbf{z}}_i = \mathrm{var}\left(\det(H_{T_k})(BA_0\mathbf{x}+B\mathbf{b}+\mathbf{d})\right)$ and $\mathbf{z} = \uplus_{i\in[m]}\widehat{\mathbf{z}}_i$.*

Proof. If $\mathtt{C}$ has no dangling variable, then we know from Lemma 3.1 that $\det(H_{\mathtt{C}}) \neq 0$; from Observation 2.8 this implies $\det(H_f) \neq 0$. Otherwise, we apply a random transformation R to $\mathbf{x}$ in f and compute the Hessian determinant of $f(R\mathbf{x})$ with respect to $\mathbf{x} \setminus \{u_0\} = \{x_1, \ldots, x_{n-1}\}$ variables. Notice that $f(R\mathbf{x}) = \mathtt{C}(BR\mathbf{x}+\mathbf{d})$. The following two claims show that this Hessian determinant is non-zero with high probability. Their proofs are given in Sections D.1 and D.2.

Claim 4.1. *The sub-matrix $[BR]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}}$ of BR, whose rows and columns are labelled by $\mathbf{x}\setminus\{u_0\}$-variables is invertible with high probability.*

Claim 4.2. *Let H_1, H_2 be the Hessians of $f(R\mathbf{x})$ and $\mathtt{C}$ with respect to $\mathbf{x}\setminus\{u_0\}$-variables, respectively. Then, $h = \det(H_1) = \beta^2 \det(H_2)(BR\mathbf{x}+\mathbf{d})$, where $\beta = \det\left([BR]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}}\right)$ and $h \neq 0$ with high probability. Also, u_0 is redundant for h with high probability.*

It is easy to see that H_2 is the Hessian of $\sum_{k\in[s']} T_k + \gamma$. Since H_2 is a block-diagonal matrix with $H_{T_k}, k \in [s']$ as the diagonal blocks, Claim 4.2 implies that $h = \beta^2 \cdot \prod_{k\in[s']} \det(H_{T_k})(BR\mathbf{x}+\mathbf{d})$. Observe that for every $k \in [s_2]$, $\det(H_{T_k})$ is non-constant. So for every $k \in [s_2]$, $\det\left(H_{T_k}\right)(BRA_0\mathbf{x}+\mathbf{d})$ is a non-constant factor of $h(A_0\mathbf{x})$. Thus, after we compute A_0' by invoking Make-Factors-Var-Disjoint() on $h(A_0\mathbf{x})$ and update A_0 to be A_0A_0' in Step 8, Claim 2.4 implies that for $k_1 \neq k_2 \in [s_2]$, $\det\left(H_{T_{k_1}}\right)(BRA_0\mathbf{x}+\mathbf{d})$ and $\det\left(H_{T_{k_2}}\right)(BRA_0\mathbf{x}+\mathbf{d})$ are variable disjoint.

For all $k \in [s]$, let $\mathbf{x}_k = \mathrm{var}(T_k)$, $h_k = \det\left(H_{T_k}\right)(BR\mathbf{x}+\mathbf{d})$, and $g_k = h_k(A_0\mathbf{x})$, where A_0 is as after Step 8. Then from Claim 2.4, g_k has no redundant variables. Let $\mathbf{z}_k = \mathrm{var}(g_k)$. Fix a $k \in [s']$. By permuting the variables of $\mathtt{C}$ if necessary, we can assume that $\mathbf{z}_k$ is also a set of essential variables for $h_k' := \det(H_{T_k})$. Let $C_k \in \mathrm{GL}(n, \mathbb{F})$ be a matrix that removes redundant variables from h_k' and $g_k' = h_k'(C_k\mathbf{x})$. Then, $\mathrm{var}(g_k') = \mathrm{var}(g_k) = \mathbf{z}_k$. Let $\mathbf{z}_k'$ be the set of truly essential variables, $\mathbf{z}_k'' = \mathbf{z}_k \setminus \mathbf{z}_k'$ be a set of ordinary essential variables, and $\mathbf{y}_k = \mathbf{x}_k \setminus \mathbf{z}_k$ be a set of redundant variables for h_k'. Let $\mathbf{y} = \uplus_{k\in[s']}\mathbf{y}_k \uplus \{u_0\}$. Note that $\mathbf{z} = \mathrm{var}\left(h(A_0\mathbf{x})\right) = \uplus_{k\in[s']}\mathbf{z}_k$, and define $\mathbf{z}' = \uplus_{k\in[s']}\mathbf{z}_k'$, $\mathbf{z}'' = \uplus_{k\in[s']}\mathbf{z}_k''$. Notice that $\mathbf{x} = \mathbf{z}' \uplus \mathbf{z}'' \uplus \mathbf{y}$. For all $x \in \mathbf{x}$ let $\ell_x^{(0)}$ be the linear form that x is mapped to by BRA_0. The following claim about the structure of BRA_0 is proved in Section D.3.

Claim 4.3 (Structure of BRA_0). *For every $k \in [s']$,*

1. *For all $z \in \mathbf{z}_k'$, $\ell_z^{(0)} \in \mathbb{F}[\mathbf{z}_k]$.*
2. *For all $z \in \mathbf{z}_k''$, $\ell_z^{(0)} = \ell_z' + \sum_{\substack{y\in\mathbf{y}_k\cap \\ \mathrm{var}\left(h_k'\right)}} \alpha_y \ell_y^{(0)}$, where $\ell_z' \in \mathbb{F}[\mathbf{z}_k]$ and for all $y \in \mathbf{y}_k \cap \mathrm{var}\left(h_k'\right)$, $\alpha_y \in \mathbb{F}$.*

Claim 3.2 implies that $\mathbf{z}_k = \mathbf{z}_k' = \mathbf{x}_k$. Hence, the above claim immediately implies that $\widehat{T}_1(A_0\mathbf{x}), \ldots, \widehat{T}_{s_1}(A_0\mathbf{x})$ are pairwise variable disjoint.

Now consider the for loop of lines 10-12. Claim 3.2 implies that for all $x \in \mathbf{x}$ connected to a $\times$ gate in $\mathtt{C}$ computing a polynomial of degree at least three, a constant multiple of the affine form $\ell_x^{(0)} + d_x$ that $BA_0\mathbf{x}+\mathbf{d}$ maps x to is in V. Also, as $N_{ess}(\mathtt{C} \mod x) < n-2$, Fact

2.4 implies that $N_{ess}\left(f(RA_0\mathbf{x}) \mod \left(\ell_x^{(0)} + d_x\right)\right) < n-2$. Conversely, if $\ell' \in V$ is such that $N_{ess}\left(f(RA_0\mathbf{x}) \mod \ell'\right) < n-2$, then it follows and from Claim 2.1 and Fact 2.4 that there exists an $x \in \mathbf{x}$ such that ℓ' is a constant multiple of $\ell_x^{(0)} + d_x$. Observe that if x is not connected to a product gate computing a polynomial of degree at least three, then $N_{ess}(\mathtt{C} \mod \alpha x) \geq n-2$ for any $\alpha \in \mathbb{F}^\times$. Hence from Fact 2.4, we have that x is connected to a product gate computing a polynomial of degree at least three. Thus the affine forms $\ell' \in V$ such that $N_{ess}\left(f(RA_0\mathbf{x}) \mod \ell'\right) < n-2$ are exactly the constant multiples of $\ell_x^{(0)} + d_x$, where x is connected to a $\times$ gate computing a polynomial of degree at least three. Because $\left\{\ell_x^{(0)} : x \in \mathbf{x}\right\}$ is linearly independent, it is possible to map all such ℓ' to distinct variables.

Observation 2.3 implies that every x connected to a $\times$ gate computing a polynomial of degree at least three is in $\mathbf{z}'$. Also, Claim 4.3 implies that if $x \in \mathbf{z}'_k$, then $\ell' \in \mathbb{F}[\mathbf{z}_k]$. Thus after the loop has been executed and A_0 updated to be RA_0C, the affine transformation $BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d}$ maps every $x \in \mathbf{x}_k$ connected to a $\times$ gate computing a polynomial of degree at least three to a constant multiple of a $\mathbf{z}_k$-variable. Also this means that $\widehat{T}_1(A_0\mathbf{x}+\mathbf{b}), \ldots, \widehat{T}_{s_1}(A_0\mathbf{x}+\mathbf{b})$ are still variable disjoint.

Immediately before Step 8 is executed, $\det(H_{T_1})(BRA_0\mathbf{x}+\mathbf{d}), \ldots, \det(H_{T_{s_2}})(BRA_0\mathbf{x}+\mathbf{d})$ are non-constant factors of $h(A_0\mathbf{x})$. So from Point 2 of Claim 2.4 there exists a partition $I_1, \ldots, I_{s_2}$ of $[m]$ such that after A_0 has been updated to be $A_0A'_0$, for all $k \in [s_2]$, $\uplus_{i\in I_k}\widehat{\mathbf{z}}_i = \mathrm{var}\left(\det(H_{T_k})(BRA_0\mathbf{x}+\mathbf{d})\right)$ and $\mathbf{z} = \uplus_{i\in[m]}\widehat{\mathbf{z}}_i$. As for all $k \in [s_2]$, C only maps some variables in $\mathbf{z}'_k$ to linear forms in $\mathbb{F}[\mathbf{z}'_k]$, we have that $\mathrm{var}\left(\det(H_{T_k})(BRA_0\mathbf{x}+\mathbf{d})\right) = \mathrm{var}\left(\det(H_{T_k})(BRA_0(C\mathbf{x}+\mathbf{b}')+\mathbf{d})\right)$. Because A_0 is updated to be RA_0C and $\mathbf{b} := RA_0\mathbf{b}'$ in Step 13, the moreover part of the lemma follows. □

Remark. After A_0 has been updated to be $RA_0A'_0$ in Step 13, for all $x \in \mathbf{x}$, we redefine $\ell_x^{(0)}$ to be the linear form that x is mapped to by BA_0. Notice that Claim 4.3 continues to hold.

4.3.2 Making bad and quadratic terms variable disjoint

The following procedure is used to make all the bad and quadratic terms variable disjoint as well. After this procedure is executed, all the non-linear terms will be variable disjoint.

Procedure 3 Make-Bad-And-Quadratic-Terms-Var-Disjoint($f(\mathbf{x})$, A_0, $\mathbf{b}$, $\mathbf{z}$, u_0)

Input. $f(\mathbf{x})$. A_0, $\mathbf{b}$, $\mathbf{z}$, and u_0 are as returned by Make-Good-Terms-Var-Disjoint($f(\mathbf{x})$).
Output. $A \in \mathrm{GL}(n,\mathbb{F})$ such that all the non-linear terms of $f(A\mathbf{x}+\mathbf{b})$ are variable disjoint.

1. $\mathbf{y} \leftarrow \mathbf{x} \setminus \mathbf{z}$.
   /* Discovering the $\mathbf{y}$ parts of quadratic forms. */
2. $\widehat{q} \leftarrow$ the degree-2 homogeneous component in $\mathbf{y}$ of $f(A_0\mathbf{x}+\mathbf{b})$ when it is viewed as a polynomial over $\mathbb{F}[\mathbf{z}]$.
3. Use sparse polynomial interpolation to interpolate $\widehat{q}$. $\{\widetilde{q}_1, \ldots, \widetilde{q}_m\} \leftarrow$ the coefficients of non-constant $\mathbf{z}$-monomials when $\widehat{q}$ is treated as a polynomial in $\mathbb{F}[\mathbf{y}]$. $\widetilde{q}_0 \leftarrow$ coefficient of 1.
4. $A'_1 \leftarrow$Make-Polys-Var-Disjoint($\widetilde{q}_0, \ldots, \widetilde{q}_m$) (see Claim 2.3). $\mathbf{u} \leftarrow \mathbf{y}$.
5. **for** $i = 0, \ldots, m$ **do**
6. $\quad p \leftarrow$ the canonical quadratic form in $\mathrm{var}\left(\widetilde{q}_i(A'_1\mathbf{x})\right)$. $C_i \leftarrow \mathrm{QFE}(\widetilde{q}_i(A'_1\mathbf{x}), p)$. Extend C_i to map every variable in $\mathbf{x} \setminus \mathrm{var}\left(\widetilde{q}_i(A'_1\mathbf{x})\right)$ to itself. $\mathbf{u} \leftarrow \mathbf{u} \setminus \mathrm{var}\left(\widetilde{q}_i(A'_1\mathbf{x})\right)$.
7. **end for**
8. $A'_1 \leftarrow A'_1 \prod_{i=0}^{m} C_i$. $A_1 \leftarrow A_0A'_1$.

/* Discovering the $\mathbf{u}$ parts of dangling linear forms. */

9. $\widehat{\ell} \leftarrow$ the degree-1 homogeneous component in $\mathbf{u}$ of $f(A_1\mathbf{x}+\mathbf{b})$ when it is viewed as a polynomial over $\mathbb{F}[\mathbf{z}]$.
10. Use sparse polynomial interpolation to interpolate $\widehat{\ell}$. Let $\mu'_1,\ldots,\mu'_{m'}$ be the non-constant $\mathbf{z}$-monomials of $\widehat{\ell}$, and $\widehat{\ell}_1\ldots,\widehat{\ell}_{m'}$ be their coefficients. $\widehat{\ell}_0 \leftarrow$ the coefficient of 1.
11. $\widehat{\ell}_{i_1},\ldots,\widehat{\ell}_{i_m} \leftarrow$ a basis of $\left\langle \widehat{\ell}_1,\ldots,\widehat{\ell}_{m'}\right\rangle$. Construct a matrix A'_2 that maps $\widehat{\ell}_{i_1},\ldots,\widehat{\ell}_{i_m}$ to distinct $\mathbf{u}$ variables, say $u_{i_1},\ldots,u_{i_m}$, such that if $\widehat{\ell}_0$ maps to u_0. Also, A'_2 acts as identity on $\mathbf{x}\setminus\mathbf{u}$.
12. $V \leftarrow \left\{(\mu'_{i_1},u_{i_1}),\ldots,(\mu'_{i_m},u_{i_m})\right\}$. $A_2 \leftarrow A_1A'_2$.
13. $C \leftarrow$ Remove-External-Vars($f(\mathbf{x}), A_2, \mathbf{b}, \mathbf{z}, \mathbf{y}, \mathbf{u}, u_0, V$) (Procedure 4).
14. $A \leftarrow A_2C$. Return A.

For now, we postpone describing the Remove-External-Vars() procedure and start the proof of correctness of the Make-Bad-And-Quadratic-Terms-Var-Disjoint() procedure. In particular, we prove the following lemma.

Lemma 4.3 (Correctness of Procedure 3)**.** *Make-Bad-And-Quadratic-Terms-Var-Disjoint($f(\mathbf{x}), A_0, \mathbf{b}, \mathbf{z}, u_0$), where $A_0, \mathbf{b}, \mathbf{z}$, and u_0 are as returned by Make-Good-Terms-Var-Disjoint($f(\mathbf{x})$) outputs an $A \in \mathrm{GL}(n,\mathbb{F})$ such that $\widehat{T}_1(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{T}_{s'}(A\mathbf{x}+\mathbf{b})$ are pairwise variable disjoint. Also, $\sum_{k=s_2+1}^{s'}\widehat{T}_k(A\mathbf{x}+\mathbf{b}) = \sum_{k=s_2+1}^{s'}(y_{k,1}+c_{k,1})(y_{k,2}+c_{k,2})$, where for all $k\in\{s_2+1,\ldots,s'\}$, $c_{k,1},c_{k,2}\in\mathbb{F}$. Further, for all $x\in\mathbf{x}$ connected to a $\times$-gate in $\mathtt{C}$ computing a polynomial of degree at least 3, $BA\mathbf{x}+B\mathbf{b}+\mathbf{d}$ maps x to constant multiple of a variable. Also, if $\mathtt{C}$ has a top dangling variable, then u_0 only appears in $\ell(A\mathbf{x}+\mathbf{b})$.*

Proof. We begin by stating the following useful claim whose proof can be found in Section D.4. For a linear form ℓ', we denote its projection to variables in a variable set $\mathbf{x}''$ by $[\ell']_{\mathbf{x}''}$. Recall that for an $x\in\mathbf{x}$, $\ell_x^{(0)}$ is the linear form that x is mapped to by BA_0.

Claim 4.4. $\left\{\left[\ell_y^{(0)}\right]_{\mathbf{y}} : y\in\mathbf{y}\right\}$ *is linearly independent.*

From Claim 3.2, only the top dangling variable, the variables in the top quadratic form, the dangling variables along skewed paths, and variables appearing in the quadratic forms along the skewed paths in $\mathtt{C}$ need not be truly essential for $\det(H_2)$. Hence, from Claim 4.3, if for some $k\in[s']$, $x\in\mathbf{x}_k$ is such that $\ell_x^{(0)}\notin\mathbb{F}[\mathbf{z}_k]$, then $k\in\{s_1+1,\ldots,s_2\}$ and x is either a dangling variable along a skewed path or a variable appearing in some quadratic form along a skewed path in T_k which is not truly essential for $\det(H_2)$, or $k\in\{s_2+1,\ldots,s'\}$ and x is a variable appearing in the top quadratic form of $\mathtt{C}$. Also, if x is a variable appearing in a skewed path in T_k, then from Lemma 4.2, $\ell_x^{(0)}+b_x=\alpha z$ for some $z\in\mathbf{z}_k$ and $\alpha\in\mathbb{F}^\times$. Suppose that the other gate connected to the parent of x is Q. Then by 'absorbing' β inside $Q(BA_0\mathbf{x}+B\mathbf{b}+\mathbf{d})$, we can assume without loss of generality that $\beta=1$.[31] Thus each skewed path is a monomial in $\mathbb{F}[\mathbf{z}]$. Also, we can assume without loss of generality that every variable appearing in a skewed path in $\mathtt{C}$ is mapped to itself by the affine transformation $BA_0\mathbf{x}+B\mathbf{b}+\mathbf{d}$, i.e. the skewed paths in $\mathtt{C}$ and $f(A_0\mathbf{x}+\mathbf{b})$ are the

[31] As mentioned in Section 4.1, absorbing β in Q means that we are starting with a different but equally valid B. This 'new' B is obtained from the 'old' B by scaling rows labelled by appropriate variables in $\mathbf{z}'$. Hence, Claim 4.3 continues to hold.

same. This is so because if a variable x appearing in a skewed path in $\mathtt{C}$ is mapped to $z \neq x$, then we can permute the variables in $\mathtt{C}$ so that the leaf labelled by x is now labelled by z.[32]

Let q_0 be the top quadratic form of $\mathtt{C}$ and $\mu_1, \ldots, \mu_m$ be all the skewed paths in $f(A_0\mathbf{x}+\mathbf{b})$ such that no variable of the quadratic forms $q_1, \ldots, q_m$ corresponding to these skewed paths in $\mathtt{C}$ appears in $\det(H_2)$ (see Claim 3.3). Also, let the corresponding quadratic forms in $f(A_0\mathbf{x}+\mathbf{b})$ be $\widehat{q}_0, \ldots, \widehat{q}_m$. Then for all $i \in \{0, \ldots, m\}$, $\widehat{q}_i = q_i(BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d})$. Suppose that $\widehat{q}_i = \ell^{(0)}_{y_{i,1,1}}\ell^{(0)}_{y_{i,1,2}} + \cdots + \ell^{(0)}_{y_{i,m_i,1}}\ell^{(0)}_{y_{i,m_i,2}}$. It follows from the discussion in the above paragraph that $\widehat{q} = \widetilde{q}_0 + \mu_1\widetilde{q}_1 + \cdots + \mu_m\widetilde{q}_m$, where $\widetilde{q}_i = \left[\ell^{(0)}_{y_{i,1,1}}\right]_{\mathbf{y}}\left[\ell^{(0)}_{y_{i,1,2}}\right]_{\mathbf{y}} + \cdots + \left[\ell^{(0)}_{y_{i,m_i,1}}\right]_{\mathbf{y}}\left[\ell^{(0)}_{y_{i,m_i,2}}\right]_{\mathbf{y}}$. Observe that each $\widetilde{q}_i$ is an n^2-sparse polynomial. As there are at most n skewed paths in $\mathtt{C}$, this means that $\widehat{q}$ is an n^3-sparse polynomial and can be interpolated efficiently. Claim 2.3 ensures that after Step 4 is executed, $\widetilde{q}_0(A'_1\mathbf{x}), \ldots, \widetilde{q}_m(A'_1\mathbf{x})$ are variable disjoint and have no redundant variables. The proof of the following claim can be found in Section D.5.

Claim 4.5. *After the for loop of lines 5-7 has been executed and A'_1 updated to be $A'_1\prod_{i=0}^{m} C_i$, for all $i \in \{0, \ldots, m\}$, $\widehat{q}_i(A'_1\mathbf{x}) = \left(y'_{i,1,1} + h_{i,1,1}\right)\left(y'_{i,1,2} + h_{i,1,2}\right) + \cdots + \left(y'_{i,m_i,1} + h_{i,m_i,1}\right)\left(y'_{i,m_i,2} + h_{i,m_i,2}\right)$, for some $y'_{i,1,1}$, $y'_{i,1,2}, \ldots, y'_{i,m_i,1}$, $y'_{i,m_i,2} \in \mathbf{y}$ and affine forms $h_{i,1,1}$, $h_{i,1,2}, \ldots, h_{i,m_i,1}$, $h_{i,m_i,2} \in \mathbb{F}[\mathbf{z}]$.*

The following observation is easy to see.

Observation 4.1. *All the $\mathbf{y}$-variables appearing in $\widehat{q}_1(A'_1\mathbf{x}), \ldots, \widehat{q}_m(A'_1\mathbf{x})$ are distinct. Also, $A'_1 \in \mathrm{GL}(n, \mathbb{F})$ and acts as identity on variables not in $\mathrm{var}(\widetilde{q}_1(A_1\mathbf{x})) \uplus \cdots \uplus \mathrm{var}(\widetilde{q}_m(A_1\mathbf{x}))$ i.e. on $\mathbf{z} \uplus \mathbf{u}$.*

In Step 8, $A_1 := A_0A'_1$. For every $k \in \{s_1+1, \ldots, s'\}$ let $\mathbf{u}_k$ be an arbitrary subset of $\mathbf{u} \setminus \{u_0\}$ of size equal to the number of dangling variables in T_k which are redundant for $\det(H_2)$. While defining $\mathbf{u}_k$s we ensure that for $k \neq k'$, $\mathbf{u}_k$ and $\mathbf{u}_{k'}$ are disjoint. Redefine $\mathbf{y}_k$ to be the union of the set of $\mathbf{y}$ variables appearing in the quadratic forms in $\widehat{T}_k(A_1\mathbf{x}+\mathbf{b})$ and $\mathbf{u}_k$. Then, by permuting the variables in $\mathtt{C}$ if necessary, we can assume that $\mathbf{y}_k \setminus \mathbf{u}_k$ are the $\mathbf{y}$ variables appearing in the quadratic forms in T_k that are redundant for $\det(H_2)$ and $\mathbf{u}_k$ are the dangling variables in T_k that are redundant for $\det(H_2)$. For all $x \in \mathbf{x}$, let $\ell^{(1)}_x$ be the linear part of the affine form that replaces x in $f(A_1\mathbf{x}+\mathbf{b})$. That is, for all $x \in \mathbf{z} \uplus \mathbf{u}$, $\ell^{(1)}_x = \ell^{(0)}_x(A'_1\mathbf{x})$ is the linear form that x is mapped to by BA_1, while for all $y \in \mathbf{y} \setminus \mathbf{u}$, if $y = y_{i,j,l}$, then $\ell^{(1)}_y = y'_{i,j,l} + h_{i,j,l}$.[33] Also by permuting the variables in $\mathtt{C}$ if required, we can assume that $y = y'_{i,j,l}$.

Claim 4.6. $\left\{\left[\ell^{(1)}_u\right]_{\mathbf{u}} : u \in \mathbf{u}\right\}$ *is linearly independent.*

A proof of the above claim can be found in Section D.6. Recall that in $f(A_0\mathbf{x}+\mathbf{b})$, $\mathbf{y}$-variables are only present in $\ell^{(0)}_x$ if x the top dangling variable, a variable in the top quadratic form, a

[32] If the permutation matrix that we need to apply to $\mathtt{C}$ is P, then the new ROF is $\mathtt{C}(P\mathbf{x})$ and the matrix transforming it to $f(A_0\mathbf{x}+\mathbf{b})$ is $P^{-1}BA_0$. While we proved Claim 4.3 for $\mathtt{C}$ and BA_0, it continues to hold for $\mathtt{C}(P\mathbf{x})$ and $P^{-1}BA_0$. This is so, because if the leaf in $\mathtt{C}$ labelled by x is labelled by x' in $\mathtt{C}(P\mathbf{x})$, then the linear form that $P^{-1}BA_0$ maps x' to, i.e., the 'new' $\ell^{(0)}_{x'}$, is the 'old' $\ell^{(0)}_x$. In other words, permuting the variables in $\mathtt{C}$ just results in the leafs of $\mathtt{C}$ and the rows of BA_0 being relabelled consistently. Through out the analysis, we shall permute the variables of $\mathtt{C}$ many times, however each time we do this, everything that we have proved up to that point for $\mathtt{C}$ and the matrix transforming it to $f(A_0\mathbf{x}+\mathbf{b})$ would continue to hold for the new $\mathtt{C}$ and the new matrix.

[33] $y'_{i,j,l} + h_{i,j,l}$ need not necessarily be $\ell^{(0)}_y(A'_1\mathbf{x})$. This is so, because for any $i \in \{0, \ldots, m\}$, an invertible matrix mapping q_i to $\widetilde{q}_i$ need not be unique.

dangling variable along a skewed path or a variable in some quadratic form along a skewed path which is not truly essential for $\det(H_2)$. Also, A'_1 acts as identity on $\mathbf{z}$, and $\left[\ell_y^{(1)}\right]_{\mathbf{y}}$ is a single variable in $\mathbf{y} \setminus \mathbf{u}$ for all $y \in \mathbf{y} \setminus \mathbf{u}$. Hence, a $\mathbf{u}$-variable is only present in $\ell_x^{(1)}$ if x is the top dangling variable (i.e. u_0) or a dangling variable along a skewed path which is not truly essential for $\det(H_2)$. So, if $u_1, \ldots, u_{m'}$ are dangling variables that are not truly essential for $\det(H_2)$ and $\mu'_1, \ldots, \mu'_m$ are the corresponding skewed paths, then $\widehat{\ell}$ in Step 10 looks like $\left[\ell_{u_0}^{(1)}\right]_{\mathbf{u}} + \mu'_1 \left[\ell_{u_1}^{(1)}\right]_{\mathbf{u}} + \cdots + \mu'_{m'} \left[\ell_{u_{m'}}^{(1)}\right]_{\mathbf{u}}$. Because $m' \leq n$, $\widehat{\ell}$ is n^2-sparse and can be interpolated efficiently. If $\mathscr{B} = \left\{\widehat{\ell}_{i_1}, \ldots, \widehat{\ell}_{i_m}\right\}$ is a basis of $\left\langle \widehat{\ell}_1, \ldots, \widehat{\ell}_{m'} \right\rangle = \left\langle \left[\ell_{u_0}^{(1)}\right]_{\mathbf{u}}, \ldots, \left[\ell_{u_{m'}}^{(1)}\right]_{\mathbf{u}} \right\rangle$, then it is clearly possible to map the linear forms $\widehat{\ell}_{i_1}, \ldots, \widehat{\ell}_{i_m}$ to distinct $\mathbf{u}$-variables. Claims 4.3 and 3.3 imply that $\left\langle \left[\ell_x^{(1)}\right]_{\mathbf{u}} : x \in \mathbf{z}''_k \uplus (\mathbf{u}_k \cap \mathrm{var}(h'_k)) \right\rangle = \left\langle \left[\ell_u^{(1)}\right]_{\mathbf{u}} : u \in \mathbf{u}_k \cap \mathrm{var}(h'_k) \right\rangle$ for every $k \in \{s_1 + 1, \ldots, s_2\}$. Thus, Claim 4.6 implies the following observation.

Observation 4.2. *For any $u \notin \mathrm{var}(\det(H_2))$, $\left[\ell_u^{(1)}\right]_{\mathbf{u}} \in \mathscr{B}$. Also, for any $k \in \{s_1 + 1, \ldots, s_2\}$, $\mathscr{B}$ contains $|\mathbf{u}_k \cap \mathrm{var}(h'_k)|$ many linear forms from $\left\{ \left[\ell_u^{(1)}\right]_{\mathbf{u}} : u \in \mathbf{u}_k \cap \mathrm{var}(h'_k) \right\}$.*

In particular, if $\mathtt{C}$ has a top dangling variable, then $\widehat{\ell}_0 = \left[\ell_{u_0}^{(1)}\right]_{\mathbf{u}} \in \mathscr{B}$ and it is mapped to u_0 in Step 11. So, after A_2 is set to $A_1 A'_2$, $\ell(A_2\mathbf{x} + \mathbf{b})$, i.e. the top linear form in $f(A_2\mathbf{x} + \mathbf{b})$ contains u_0. Also, because A'_1 acts as identity on $\mathbf{z} \uplus \mathbf{u}$ and A'_2 act as identity on $\mathbf{z} \uplus \mathbf{y}$, $A'_1 A'_2$ is identity on $\mathbf{z}$. For every $k \in \{s_1 + 1, \ldots, s_2\}$, let $\mathbf{x}'_k \subseteq \mathbf{z}''_k \uplus \mathbf{u}_k$, $|\mathbf{x}'_k| = |\mathbf{u}_k|$ be such that $\left\{ \left[\ell_x^{(1)}\right]_{\mathbf{u}} : x \in \mathbf{x}'_k \right\} \subseteq \mathscr{B}$. Let $\mathbf{x}' = \uplus_{s_1+1 \leq k \leq s_2} \mathbf{x}'_k \uplus \{u_0\}$. Then the following observation is easy to see.

Observation 4.3. *For all $x \in \mathbf{x}'$, $\ell_x^{(1)}(A'_2\mathbf{x})$ looks like $u + h'_u$, for some $u \in \mathbf{u}$ and $h'_u \in \mathbb{F}[\mathbf{z}, \mathbf{y} \setminus \mathbf{u}]$. Also, if the skewed path corresponding to x is μ, then $(\mu, u) \in V$.*

We now show that $\mathbf{x}'$ is in a set of redundant variables for $\det(H_2)$.

Claim 4.7. *$\mathbf{x}' \uplus (\mathbf{y} \setminus \mathbf{u})$ is a set of redundant variables for $\det(H_2)$.*

The above claim is proved in Section D.7. Using an argument similar to the one used to prove Claim 4.3, we can prove the following observation.

Observation 4.4. *For all $k \in \{s_1 + 1, \ldots, s_2\}$ and all $x \in \mathbf{z}''_k \uplus (\mathbf{u}_k \cap \mathrm{var}(h'_k))$, $\ell_x^{(1)} = \ell''_x + \sum_{\substack{x' \in \mathbf{x}' \cap \\ \mathrm{var}(h'_k)}} \alpha'_{x'} \ell_{x'}^{(1)}$, where $\ell''_x \in \mathbb{F}[\mathbf{z}_k]$ and $\alpha'_{x'} \in \mathbb{F}$ for all $x' \in \mathbf{x}'_k \cap \mathrm{var}(h'_k)$.*

For all $x \in \mathbf{x}$, let $\ell_x^{(2)} = \ell_x^{(1)}(A'_2\mathbf{x})$ be the linear part of the affine from that replaces x in $f(A_2\mathbf{x} + \mathbf{b})$. Because $A'_1 A'_2$ acts as identity on $\mathbf{z}_k$, Observations 4.2 and 4.4 imply that if for all $k \in \{s_1 + 1, \ldots, s_2\}$ and $x \in \mathbf{x}'_k$, we can remove variables not in $\mathbf{z}_k \uplus \mathbf{y}_k$ - i.e. "external" variables - from $\ell_x^{(2)}$, then all linear forms corresponding to dangling variables along skewed paths in T_k would just be in $\mathbf{z}_k \uplus \mathbf{y}_k$ variables. Similarly, because A'_2 acts as identity on $\mathbf{y}$, Claim 4.5 implies that if we can remove variables not in $\mathbf{z}_k$ from $\ell_y^{(2)}$ for all $y \in \mathbf{y}_k \setminus \mathbf{u}_k$, then all linear forms corresponding to variables appearing in quadratic forms along skewed paths in T_k would just be in $\mathbf{z}_k \uplus \mathbf{y}_k$ variables. Then all the non-linear terms of $f(A_2\mathbf{x} + \mathbf{b})$ would become variable disjoint. We now describe the Remove-External-Vars() procedure and show that it does just this.

Procedure 4 Remove-External-Vars($f(\mathbf{x}), A_2, \mathbf{b}, \mathbf{z}, \mathbf{y}, \mathbf{u}, u_0, V$).

Input. $f(\mathbf{x})$, A_2, $\mathbf{b}$, $\mathbf{z}$, $\mathbf{y}$, $\mathbf{u}$, u_0, and V are as in Step 12 of Procedure 3.
Output. $A \in \mathrm{GL}(n, \mathbb{F})$ such that all the non-linear terms of $f(A\mathbf{x} + \mathbf{b})$ are variable disjoint.

/* Removing external $\mathbf{z}$-variables from quadratic forms and external $\mathbf{y}$-variables from linear forms. */

1. $A'_3 \leftarrow I_{n\times n}$.
2. **for** $y \in \mathbf{y} \setminus \mathbf{u}$ **do**
3. Interpolate $g \leftarrow \frac{\partial f(A_2\mathbf{x}+\mathbf{b})}{\partial y}$. If μ is the $\mathbf{z}$-monomial multiplied to the only $\mathbf{y}$-variable, say y', in g, then write $g = \mu(y' + \ell'_{y'} + \alpha_{y'}) + r(\mathbf{z})$, where $\ell'_{y'} \in \mathbb{F}[\mathbf{z}]$ is a linear form, $\alpha_{y'} \in \mathbb{F}$, and $r(\mathbf{z}) \in \mathbb{F}[\mathbf{z}]$.
4. Update A'_3 so that it maps y' to $y' - \ell'_{y'}$.
5. **for** every monomial μ' in $r(\mathbf{z})$ **do**
6. If there exists a u' such that $(\mu', u') \in V$, update A'_3 to map u' to $u' - \beta y$, where β is the coefficient of μ' in $r(\mathbf{z})$.
7. **end for**
8. **end for**
9. $A_3 \leftarrow A_2 A'_3$.

/* Removing external $\mathbf{z}$ variables from linear forms. */

10. $A'_4 \leftarrow I_{n\times n}$. $F \leftarrow$ a subset of $\mathbb{F}$ of size n^5.
11. **for** $(\mu, u) \in V$ such that $\deg(\mu) \geq 2$ **do**
12. **for** $z \in \mathbf{z}$ **do**
13. $g \leftarrow f(A_3(\mathrm{var}(\mu),\ z,\ \mathbf{x} \setminus (\mathrm{var}(\mu) \uplus \{z\}) = \mathbf{0}) + \mathbf{b})$.
14. Interpolate $\frac{\partial g}{\partial z}$. Let α be the coefficient of μ in $\frac{\partial g}{\partial z}$. Update A'_4 to map u to $u - \alpha z$.
15. **end for**
16. **end for**
17. **for** $(\mu, u) \in V$ such that $\deg(\mu) = 1$ **do**
18. **for** $z \in \mathbf{z}$ **do**
19. Set all variables in $f(A_3 A'_4 \mathbf{x} + \mathbf{b})$ other then $\mathrm{var}(\mu)$ and z to random elements from F. $g \leftarrow$ the resulting polynomial.
20. Interpolate $\frac{\partial g}{\partial z}$. Let α be the coefficient of μ in $\frac{\partial g}{\partial z}$. Update A'_4 to map u to $u - \alpha z$.
21. **end for**
22. **end for**
23. $A \leftarrow A_3 A'_4$. Return A.

We first consider the for loop of lines 2-8 and show that the matrix A'_3 computed by this loop is such that it removes all variables not in $\mathbf{z}_k \uplus \mathbf{y}_k$ from $\ell_y^{(2)}$ for all $y \in \mathbf{y}_k$ and $k \in \{s_1 + 1, \ldots, s'\}$, and removes all variables in $\mathbf{y} \setminus \mathbf{y}_k$ from $\ell_x^{(2)}$ for all $x \in \mathbf{x}'_k$ and $k \in \{s_1 + 1, \ldots, s_2\}$. Before arguing this, we remark that in any iteration of this loop, g is a $2n$-sparse polynomial. This is so, because any $y \in \mathbf{y}$ is only present in $\ell_y^{(2)}$ and in $\ell_x^{(2)}$ for $x \in \mathbf{x}'$. Thus, every monomial in $r(\mathbf{z})$ is a skewed

path and there are at most n skewed paths. The following claim is proved in Section D.8.

Claim 4.8. *The matrix A'_3 computed after the execution of the for loop of lines 2-8 is such that for every $k \in \{s_1+1,\ldots,s_2\}$ and every $y' \in \mathbf{y}_k \setminus \mathbf{u}_k$, $\ell^{(2)}_{y'}(A'_3\mathbf{x}) \in \mathbb{F}[\mathbf{z}_k \uplus \mathbf{y}_k]$.*

To show that $BA_2A'_3$ maps the top quadratic form to $\sum_{k=s_2+1}^{s'}(y_{k,1}+c_{k,1})(y_{k,2}+c_{k,2})$ and A'_3 removes external $\mathbf{y}$ variables from $\ell^{(2)}_{u_0}$, we shall use the following two observations.

Observation 4.5. *For any $y \in \uplus_{k'=s_2+1}^{s'}\mathbf{y}_{k'}$ and any $k \in \{s_1+1,\ldots,s_2\}$, if $T_k = zQ$ and the top dangling variable of Q is x, then it can be assumed without loss of generality that y is not present in $\ell^{(2)}_x$.*

Proof. Suppose that yy' is a term in $\sum_{k'=s_2+1}^{s'} T_{k'}$ and that the coefficient of y in $\ell^{(2)}_x$ is β. Then we 'absorb' βz in $\ell^{(2)}_{y'}$ and subtract $\beta\left(\ell^{(2)}_y + c - y\right)$ from $\ell^{(2)}_x$; here c is the constant term of the affine form that replaces y in $f(A_2\mathbf{x}+\mathbf{b})$. This does not change $f(A_2\mathbf{x}+\mathbf{b})$. □

Observation 4.6. *If $\mathtt{C}$ has a top dangling variable, then for any $y \in \uplus_{k=s_2+1}^{s'}\mathbf{y}_k$, it can be assumed without loss of generality that the affine form replacing y in $f(A_2\mathbf{x}+\mathbf{b})$ has no constant.*

Proof. Suppose that $T_k = y_1y_2$ for some $k \in \{s_2+1,\ldots,s'\}$ and that $\widehat{T}_k(A_2\mathbf{x}+\mathbf{b}) = (\ell^{(2)}_{y_1}+c_1)(\ell^{(2)}_{y_2}+c_2)$, where $c_1, c_2 \in \mathbb{F}$. Then we add $c_2\ell^{(2)}_{y_1} + c_1\ell^{(2)}_{y_2}$ to $\ell^{(2)}_{u_0}$ and add c_1c_2 to the constant of the affine form replacing u_0 in $f(A_2\mathbf{x}+\mathbf{b})$. This does not change $f(A_2\mathbf{x}+\mathbf{b})$. □

We call every $x \in \mathbf{x}'$ such that some $T_k = zQ$ and x is the top dangling variable of Q, a *bad dangling variable*. For every $y \in \uplus_{k'=s_2+1}^{s'}\mathbf{y}_{k'}$, every bad dangling variable x, and u_0 we redefine $\ell^{(2)}_y$, $\ell^{(2)}_x$ and $\ell^{(2)}_{u_0}$ as mentioned in the proofs of the above observations.

Claim 4.9. *The matrix A'_3 computed after the execution of the for loop of lines 2-8 is such that for every $y' \in \uplus_{s_2+1\le k\le s'}\mathbf{y}_k$, $\ell^{(2)}_{y'}(A'_3\mathbf{x}) = y' + c$ for some $c \in \mathbb{F}$.*

A proof of the above claim can be found in Section D.9. Claims 4.8 and 4.9 ensure that external variables are removed from the linear forms corresponding to variables appearing in quadratic forms along skewed paths and the top quadratic form. The following claim proves that external $\mathbf{y}$ variables are removed from linear forms corresponding to dangling variables along skewed paths and the top dangling variable. It is proved in Section D.10.

Claim 4.10. *The matrix A'_3 computed after the execution of the for loop of lines 2-8 is such that for every $k \in \{s_1+1,\ldots,s_2\}$ and $x \in \mathbf{x}'_k$, $\ell^{(2)}_x(A'_3\mathbf{x})$ does not contain any variable from $\mathbf{y} \setminus \mathbf{y}_k$. Also if $\mathtt{C}$ has a top dangling variable, then $\ell^{(2)}_{u_0}(A'_3\mathbf{x})$ does not contain any $\mathbf{y}$ variable other than u_0.*

After A_3 has been defined as $A_2A'_3$, let $\ell^{(3)}_x$ be the linear part of the affine form replacing x in $f(A_3\mathbf{x}+\mathbf{b})$. Note that for all x other than those in $\uplus_{s_2+1\le k\le s'}\mathbf{y}_k$, bad dangling variables, and u_0, $\ell^{(3)}_x = \ell^{(2)}_x(A'_3\mathbf{x})$. Now from Claim 4.8, for all $k \in \{s_1+1,\ldots,s_2\}$, the only $x \in \mathbf{x}_k$ for which $\ell^{(3)}_x$ contains variables not in $\mathbf{z}_k \uplus \mathbf{y}_k$ are dangling variables along skewed paths. Also, for all $x \in \mathbf{x}'_k$, Claim 4.10 implies that $\ell^{(3)}_x \in \mathbb{F}[\mathbf{z} \uplus \mathbf{y}_k]$. Now $A'_2A'_3$ acts as identity on $\mathbf{z}$ and Claim 3.4 implies that no bad dangling variable is in $\mathrm{var}(\det(H_2))$. Thus Observation 4.4 implies that for all $k \in \{s_1+1,\ldots,s_2\}$, $x \in \mathbf{z}''_k \uplus \mathbf{u}_k$, $\ell^{(3)}_x \in \mathbb{F}[\mathbf{z} \uplus \mathbf{y}_k]$. The following claim is proved in Section D.11

Claim 4.11. *The matrix A_4' computed after the execution of the for loop of lines 11-16 is such that for every $k \in \{s_1+1, \ldots, s_2\}$, $x \in \mathbf{x}'_k$ is not a bad dangling variable, $\ell_x^{(3)}(A_4'\mathbf{x}) \in \mathbb{F}[\mathbf{z}_k \uplus \mathbf{y}_k]$.*

The above claim immediately implies that for all $k \in \{s_1+1, \ldots, s_2\}$, $\ell_x^{(3)}(A_4'\mathbf{x}) \in \mathbb{F}[\mathbf{z}_k \uplus \mathbf{y}_k]$ for all $x \in \mathbf{z}''_k \uplus \mathbf{u}_k$ which is not a bad dangling variable. To prove an analogous statement for the bad dangling variables we need the following observation. Note that Claim 3.4 and Observation 4.2 imply that every bad dangling variable is in $\mathbf{x}'$.

Observation 4.7. *Suppose that $x_1, \ldots, x_m$ are all the bad dangling variables, the corresponding skewed paths are $\mu_1, \ldots, \mu_m$, the sole $\mathbf{u}$ variables in $\ell_{x_1}^{(3)}, \ldots, \ell_{x_m}^{(3)}$ are $u_1, \ldots, u_m$, and the for loop of lines 17-22 processes $(\mu_1, u_1), \ldots, (\mu_m, u_m)$ in that order. Then, it can be assumed without loss of generality that for all $i \in [m]$ and all $j < i$, $\ell_{x_i}^{(3)}$ does not contain z_j.*

Proof. For all $i \in [m]$ and all $j < i$, let the coefficient of z_j in $\ell_{x_i}^{(3)}$ be $\beta_{i,j}$. For all $j \in [m]$, we 'absorb' $\sum_{i=j+1}^{m} \beta_{i,j} z_i$ in $\ell_{x_j}^{(3)}$ and remove $\beta_{i,j} z_j$ from $\ell_{x_i}^{(3)}$ for all $i > j$. This does not change $f(A_3\mathbf{x} + \mathbf{b})$. □

For every bad dangling variable x, we redefine $\ell_x^{(3)}$ as mentioned in the proof of the above observation. The following claim shows that variables in $\mathbf{z} \setminus \mathbf{z}_k$ are removed from $\ell_x^{(3)}$ for every bad dangling variable $x \in \mathbf{x}'_k$ as well. It is proved in Section D.12.

Claim 4.12. *The matrix A_4' computed after the execution of the for loop of lines 17-22 is such that for every $k \in \{s_1+1, \ldots, s_2\}$, and every bad dangling variable $x \in \mathbf{x}'_k$, $\ell_x^{(3)}(A_4') \in \mathbb{F}[\mathbf{z}_k \uplus \mathbf{y}_k]$.*

Let $\ell_x^{(4)}$ be the linear part of the affine form replacing x in $f(A\mathbf{x} + \mathbf{b})$. For all $k \in [s']$ and $x \in \mathbf{x}_k$, $\ell_x^{(4)}$ is now a linear form in $\mathbf{z}_k \uplus \mathbf{y}_k = \mathbf{x}_k$. Also, Claim 4.9 and the fact that A_4' acts as identity on $\mathbf{y}$ implies that $\sum_{k \in s_2+1}^{s'} \widehat{T}_k(A\mathbf{x} + \mathbf{b}) = \sum_{k \in s_2+1}^{s'} (y_{k,1} + c_{k,1})(y_{k,2} + c_{k,2})$. Now, as seen in the proof of Lemma 4.2, for every $x \in \mathbf{x}$ connected to a $\times$ gate computing a polynomial of degree at least three, $\ell_x^{(1)}$ is a constant multiple of a $\mathbf{z}$-variable. As $A_1' \cdots A_4'$ acts as identity on $\mathbf{z}$, so is $\ell_x^{(4)}$. Moreover, if $\mathtt{C}$ has a top dangling variable $u_0 = x_n$, then from Claim 4.10, the only $\mathbf{u}$ variable in $\ell_{u_0}^{(2)}$ was u_0. As A_3' merely translates u_0 by constant multiples of $\mathbf{y}$-variables and A_4' acts as identity on u_0, the only $\mathbf{u}$ variable in $\ell_{u_0}^{(4)}$ is u_0. Also, $u_0 \notin \text{var}\left(\ell_x^{(4)}\right)$ for any $x \neq u_0$. □

4.3.3 Discovering the top linear form

We begin by stating the following useful claim whose proof can be found in Section D.13.

Claim 4.13 (Learning variable sets)**.** *After Step 11 of Algorithm 1 is executed, $\mathbf{z}_k = \text{var}\left(\widehat{T}_k(A\mathbf{x} + \mathbf{b})\right)$ for all $k \in [s_2]$.*[34]

[34] Here we are overloading the notation. Now $\mathbf{z}_k = \text{var}\left(\widehat{T}_k(A\mathbf{x} + \mathbf{b})\right)$. but in Sections 4.3.1 and 4.3.2 it was a set of essential variables of $\det(H_{T_k})$ evaluated at $BRA\mathbf{x} + \mathbf{d}$. The new $\mathbf{z}_k$ is the union of the old $\mathbf{z}_k$ and $\mathbf{z}_y$.

Because of Step 12 of Algorithm 1, the following procedure will only be called if $\mathtt{C}$ has a top dangling variable. It finds an affine form ℓ' such that when we map u_0 to $u_0 - \ell'$ in $f(A\mathbf{x}+\mathbf{b})$, all its terms become variable disjoint and $\ell(A\mathbf{x}+\mathbf{b})$ becomes $u_0 + c$ for some $c \in \mathbb{F}$ (recall that ℓ is the affine form that the top dangling variable is mapped to by $B\mathbf{x}+\mathbf{d}$). This is done in s_2 iterations. In the k-th iteration it finds ℓ' restricted to $\mathbf{z}_k$ variables, denoted by ℓ_k.

Procedure 5 Find-Top-Linear-Form(f')

Input: $f' = f(A\mathbf{x}+\mathbf{b})$, where A and $\mathbf{b}$ as after Step 3 of Algorithm 1.
Output: An affine form ℓ' such that all terms in $f'\left(\mathbf{x} \setminus \{u_0\}, u_0 = u_0 - \ell'\right)$ are variable disjoint.

1. **for** $k \in [s_2]$ **do**
2. $\quad \widehat{T} \leftarrow f'(\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k = \mathbf{0})$. $h' \leftarrow$ the Hessian determinant of $\widehat{T}$ with respect to $\mathbf{z}_k$-variables. $N \leftarrow$ the set of irreducible factors of h'. $F \leftarrow$ a subset of $\mathbb{F}$ of size at least n^5.
3. $\quad$ **for** $\widehat{Q} \in N$ **do**
4. $\quad\quad$ **if** $\widehat{Q}$ is not linear **then**

/* $\widehat{Q}$ is non-linear. */

5. $\quad\quad\quad \mathbf{a}_1, \ldots, \mathbf{a}_{|\mathbf{z}_k|} \leftarrow$ vectors of size $|\mathbf{z}_k|$ containing random elements from F. $t \leftarrow$ a fresh variable.
6. $\quad\quad\quad \forall i \in [|\mathbf{z}_k|]$, interpolate $\widehat{Q}(t\mathbf{a}_i)$ and $\widehat{T}(t\mathbf{a}_i)$. Discover $\widehat{Q}'_i(t)$ and $\beta_{i,0},\ \beta_{i,1} \in \mathbb{F}$ such that $\widehat{Q}(t\mathbf{a}_i) \cdot \widehat{Q}'_i(t) + \beta_{i,1} \cdot t + \beta_{i,0} = \widehat{T}(t\mathbf{a}_i)$ by solving a system of linear equations in the coefficients of $\widehat{Q}'_i(t)$ and $\beta_{i,0},\ \beta_{i,1}$.
7. $\quad\quad\quad$ Interpolate $\sum_{z \in \mathbf{z}_k} \alpha_z z$ using $\beta_{i,1}, \ldots, \beta_{|\mathbf{z}_k|,1}$. If $\widehat{T} - \sum_{z \in \mathbf{z}_k} \alpha_z z - \beta_{1,0}$ is reducible, $\ell_k \leftarrow \sum_{z \in \mathbf{z}_k} \alpha_z z - \beta_{1,0}$ and $\widehat{T} \leftarrow \widehat{T} - \ell_k$. Break.
8. $\quad\quad$ **else**

/* $\widehat{Q}$ is linear. */

9. $\quad\quad\quad$ Suppose $\widehat{Q} = z$; if not, move to the next iteration. $\mathbf{a}_i, \ldots, \mathbf{a}_{|\mathbf{z}_k|-1} \leftarrow$ vectors of size $|\mathbf{z}_k| - 1$ containing random elements from F. $t \leftarrow$ a fresh variable.
10. $\quad\quad\quad \forall i \in [|\mathbf{z}_k| - 1]$, interpolate $\widehat{T}(z, \mathbf{z}_k \setminus \{z\} = t\mathbf{a}_i)$. Find $\widehat{Q}'_i(z,t)$ and $\beta_{i,0},\ \beta_{i,1},\ \beta_{i,2} \in \mathbb{F}$ such that $z \cdot \widehat{Q}'_i(z,t) + \beta_{i,2} \cdot z + \beta_{i,1} \cdot t + \beta_{i,0} = \widehat{T}(z, \mathbf{z}_k \setminus \{z\} = t\mathbf{a}_i)$ by solving a system of linear equations in the coefficients of $\widehat{Q}'_i(z,t)$ and $\beta_{i,0},\ \beta_{i,1},\ \beta_{i,2}$.
11. $\quad\quad\quad$ Interpolate $\sum_{z' \in \mathbf{z}_k \setminus \{z\}} \alpha_{z'} z'$ using $\beta_{i,1}, \ldots, \beta_{|\mathbf{z}_k|-1,1}$. If $\widehat{T} - \sum_{z' \in \mathbf{z}_k \setminus \{z\}} \alpha_{z'} z' - \beta_{1,0}$ is reducible, $\ell_k \leftarrow \sum_{z' \in \mathbf{z}_k \setminus \{z\}} \alpha_{z'} z' - \beta_{1,0}$ and $\widehat{T} \leftarrow \widehat{T} - \ell_k$. Break.
12. $\quad\quad$ **end if**
13. $\quad$ **end for**
14. **end for**
15. $\ell' \leftarrow \sum_{k \in [s_2]} \ell_k$. Return ℓ'.

We now prove the following lemma.

Lemma 4.4 (Correctness of Procedure 5)**.** *Find-Top-Linear-Form($f(A\mathbf{x}+\mathbf{b})$), where A and $\mathbf{b}$ are as after Step 3 of Algorithm 1, finds an affine form ℓ' such that when u_0 is mapped to $u_0 - \ell'$ in $f(A\mathbf{x}+\mathbf{b})$, all its terms become variable disjoint and $\ell(A\mathbf{x}+\mathbf{b})$ becomes $u_0 + \alpha$ for some $\alpha \in \mathbb{F}$.*

Proof. Because of Step 12 of Algorithm 1, this procedure will only be called if $\mathtt{C}$ has a top dangling variable. Recall that the variable sets of $\widehat{T}_1(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{T}_{s_2}(A\mathbf{x}+\mathbf{b})$, and $\sum_{k=s_2+1}^{s'}\widehat{T}_k(A\mathbf{x}+\mathbf{b})$ are $\mathbf{z}_1,\ldots,\mathbf{z}_{s_2}$, and $\mathbf{y}\setminus\{u_0\}$ respectively. From Observation 4.3 the coefficient of u_0 in $\ell(A\mathbf{x}+\mathbf{b})$ is 1, and from Claim 4.10 no $y\in\mathbf{y}\setminus\{u_0\}$ appears in $\ell(A\mathbf{x}+\mathbf{b})$. Let $\ell(A\mathbf{x}+\mathbf{b})=u_0+\sum_{z\in\mathbf{z}}c_z\cdot z+c_0$; recall that $\mathbf{z}=\mathbf{z}_1\uplus\cdots\uplus\mathbf{z}_{s_2}$. Fix a $k\in[s_2]$. We now show that in k-th iteration of the loop of lines 1-14 (the outer loop), the procedure finds ℓ_k which is ℓ' restricted to $\mathbf{z}_k$ variables. Towards this, we first show that in the k-th iteration of the outer loop, $\widehat{T}$ is reducible after the execution of the for loop of lines 3-13 (the inner loop).

Claim 4.14. *For any $k\in[s_2]$, after the execution of the inner loop during the k-th iteration of the outer loop, $\widehat{T}$ is reducible.*

Proof. At the beginning of the k-th iteration, $\widehat{T}=\widehat{T}_k(A\mathbf{x}+\mathbf{b})+[\ell(A\mathbf{x}+\mathbf{b})]_{\mathbf{z}_k}+\gamma'$, for some $\gamma'\in\mathbb{F}$. In the procedure N is the set of irreducible factors of h' which is the Hessian determinant of $\widehat{T}$ with respect to the $\mathbf{z}_k$-variables. Let $\widehat{T}_k=\widehat{Q}_{k,1}\cdots\widehat{Q}_{k,m_k}$. It follows from Corollary 3.1 and Fact 2.8, that at least one of the $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{Q}_{k,m_k}(A\mathbf{x}+\mathbf{b})$ is in an irreducible factor of h'. Hence, a constant multiple of at least one of the $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{Q}_{k,m_k}(A\mathbf{x}+\mathbf{b})$ is present in N along with some other 'bad' factors. Fix a $\widehat{Q}\in N$. Then it is either a 'good' non-linear factor, 'good' linear factor, or a bad factor. In the following two claims we show that in the first two cases, $\widehat{T}$ is made reducible.

Claim 4.15. *If $\widehat{Q}$ is a constant multiple of one of the $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{Q}_{k,m_k}(A\mathbf{x}+\mathbf{b})$ and is non-linear, then after the execution of the first inner loop, $\widehat{T}$ is reducible.*

Claim 4.16. *If $\widehat{Q}$ is a constant multiple of one of the $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{Q}_{k,m_k}(A\mathbf{x}+\mathbf{b})$ and is linear, after the execution of the first inner loop, $\widehat{T}$ is reducible.*

The above claims are proved in Sections D.14 and D.15, respectively. Consider an iteration of the inner loop for a bad $\widehat{Q}$. If in this iteration $\widehat{T}$ is made reducible, then there is nothing to prove. Otherwise it follows from the above two observations that for all previous iterations of the inner loop, $\widehat{Q}$ must have been a bad factor. It follows from Corollary 3.1, that at least one of $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{Q}_{k,m_k}(A\mathbf{x}+\mathbf{b})$ is an irreducible factor of h'. This means that the next iteration of this loop will be executed and this will continue to happen until for an iteration $\widehat{Q}$ is a constant multiple of $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}),\ldots,\widehat{Q}_{k,m_k}(A\mathbf{x}+\mathbf{b})$. The claim follows from Claims 4.15 and 4.16. □

The proof of the following structural result can be found in Section D.16.

Claim 4.17. *Suppose that $Q_1\cdots Q_m+u$ is a canonical ROF. Let $T=Q_1\cdots Q_m+\ell(\mathbf{z})$, where ℓ is a non-zero affine form and $Q_1\cdots Q_m$ is not a quadratic polynomial. Then T is reducible if and only if for some $l\in[m]$, Q_l is an affine form in a single variable and ℓ is a constant multiple of Q_l.*

We complete the proof of the lemma by combining Claims 4.14 and 4.17. As the k-th iteration of the outer loop only works with $\mathbf{z}_k$, we can analyse each iteration in isolation. Claim 4.14 implies that after the execution of the inner loop, $\widehat{T}$ is reducible. Initially, $\widehat{T}=\widehat{T}_k(A\mathbf{x}+\mathbf{b})+[\ell(A\mathbf{x}+\mathbf{b})]_{\mathbf{z}_k}+\gamma'$ for some $\gamma'\in\mathbb{F}$. The inner loop only subtracts an affine form from $\widehat{T}$. So after the execution of the inner loop, $\widehat{T}=\widehat{T}_k(A\mathbf{x}+\mathbf{b})+\widetilde{\ell}$ for some affine form $\widetilde{\ell}$. Notice that $\widehat{T}\in\mathrm{orb}(T_k+\widetilde{\ell}')$ for $\widetilde{\ell}':=\widetilde{\ell}(A^{-1}B^{-1}(\mathbf{x}-B\mathbf{b}-\mathbf{d}))$. Let $T_k=Q_{k,1}\ldots Q_{k,m_k}$, $\widehat{T}_k=\widehat{Q}_{k,1}\ldots\widehat{Q}_{k,m_k}$, and $\widehat{Q}_{k,l}=Q_{k,l}(B\mathbf{x}+\mathbf{d})$ for all $l\in[m_k]$. Claim 4.14 implies that $\widehat{T}$ is reducible. Then, if none of the factors $\widehat{Q}_{k,1},\ldots,\widehat{Q}_{k,m_k}$ are linear, Claim 4.17 implies that $\widetilde{\ell}=0$. On the other hand, if one of the factors, say $\widehat{Q}_{k,1}$ is linear,

then Claim 4.17 implies that $\widetilde{\ell} = c'_k \cdot \widehat{Q}_{k,1}$ for some $c'_k \in \mathbb{F}$ and $\widehat{T} = \widehat{Q}_{k,1}\left(\widehat{Q}_{k,2}\dots\widehat{Q}_{k,m_k} + c'_k\right)$. In the first case, ℓ_k must be $[\ell(A\mathbf{x})]_{\mathbf{z}_k} + \gamma'$. Because $Q_{k,1}$ is a variable connected to a $\times$ gate computing a polynomial of degree at least 3, Lemma 4.3 implies that $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b})$ is a constant multiple of a variable, say z. Thus, in this case, ℓ_k and $[\ell]_{\mathbf{z}_k}$ must agree on the coefficients of all $z' \in \mathbf{z}_k$ except perhaps that of z. For every $k \in \{s_2+1,\dots,s'\}$, every $k \in [s_2]$ such that T_k is in the first case, and every $k \in [s_2]$ in the second case that looks like xQ, where Q has a top-dangling y, let $T'_k = T_k$. For every other k, let $T'_k = Q_{k,1}\left(Q_{k,2}\dots Q_{k,m_k} + c'_k\right)$. We also redefine $\mathbf{d}$ as follows: For every $k \in [s_2]$ such that T_k is in the second case, it looks like zQ, and the top dangling variable of Q is y, we add c'_k to the y-th entry of $\mathbf{d}$; all other entries remain unchanged. If we redefine $\widehat{T}(A\mathbf{x}+\mathbf{b}) = T'_k(BA\mathbf{x}+B\mathbf{b}+\mathbf{d})$, and $\mathtt{C}' = T'_1+\dots+T'_s+\gamma$, then $f(A\mathbf{x}+\mathbf{b}) = \widehat{T}_1+\dots+\widehat{T}_s+\gamma = \mathtt{C}'(BA\mathbf{x}+B\mathbf{b}+\mathbf{d})$. Now, when we map u_0 to $u_0-\ell'$ in $f(A\mathbf{x}+\mathbf{b})$, all its terms are variable disjoint and $\ell(A\mathbf{x}+\mathbf{b})$ becomes $u_0+\alpha$ for some $\alpha \in \mathbb{F}$. □

Remark. Notice that $\mathtt{C}'$ need not be a canonical ROF. However, for all $k \in [s_2]$, all the factors of T'_k are still canonical. As we only recursively perform equivalence test on the factors of $\widehat{T}_k(A\mathbf{x})$, $\mathtt{C}'$ not being canonical is not a problem.

4.3.4 Obtaining efficient black-box access to a term

The next procedure is used to obtain black-box access to a term $\widehat{T}_k(A\mathbf{x})$ using *a single* query to f.

Procedure 6 Compute-Term-Black-Box(g)

Input: Black-box access to a term of $f(A\mathbf{x})$ plus an unknown constant.
Output: Black-box access to the term using just one query to the black-box of f.

1. $F \leftarrow$ a subset of $\mathbb{F}$ of size at least n^5.
2. Obtain black-box access to $\det(H_g)$ with respect to $\mathrm{var}(g)$ and factorize it using the algorithm in [KT90]. $N \leftarrow$ set of black-boxes of the irreducible factors.
3. **for** $r \in N$ **do**
4. $\quad\mathbf{a} \leftarrow$ a vector of size $|\mathrm{var}(g)|$ containing random elements from F. For a fresh variable t, interpolate $r(t\mathbf{a})$ and $g(t\mathbf{a})$.
5. $\quad$Discover $r'(t)$ and $\beta \in \mathbb{F}$ such that $r(t\mathbf{a})r'(t)+\beta = g(t\mathbf{a})$ by solving a system of linear equations in the coefficients of r' and β.
6. $\quad$If $g-\beta$ is reducible, then return black-box access to $g-\beta$.
7. **end for**

The following claim proved in Section D.17 establishes the correctness of the above procedure.

Lemma 4.5 (Correctness of Procedure 6)**.** *Compute-Term-Black-Box*$(f\,(A\,(\mathbf{z}_k, \mathbf{x}\setminus\mathbf{z}_k = \mathbf{0})))$ *gives black-box access to* $\widehat{T}_k(A\mathbf{x})$ *with high probability. Also, one query to* $\widehat{T}_k(A\mathbf{x})$ *needs just one query to* f*.*

4.3.5 Proof of Lemma 4.1

By induction on the product-depth Δ of $\mathtt{C}$. If $\Delta = 0$, as $\mathtt{C}$ is a canonical ROF, $\mathtt{C} = x_1$ and f is an affine form. Since all variables in f are essential, $n = 1$ and $f = \alpha_1 x_1 + \alpha_0$ for some $\alpha_0, \alpha_1 \in \mathbb{F}$, $\alpha_1 \neq 0$. Then, $f(I_{n\times n}\mathbf{x}) \in$ PS-orb($\mathtt{C}$) and the algorithm works correctly for product-depth 0 ROFs.

Assume that the algorithm works correctly for all polynomials in the orbit of a canonical ROF of product-depth $\Delta \geq 0$ and let $\mathtt{C}$ be a canonical ROF of product-depth $\Delta + 1$. Recall that the algorithm is given black-box access to an $f \in \mathbb{F}[\mathbf{x}]$ such that there exist a $B \in \mathrm{GL}(n, \mathbb{F})$ and a $\mathbf{d} \in \mathbb{F}^n$ satisfying $f = \mathtt{C}(B\mathbf{x} + \mathbf{d})$. Also, there are no redundant variables in f. Further $\mathtt{C} = T_1 + \cdots + T_s + \gamma$, where $T_1, \cdots, T_s$ are $\times$-rooted canonical ROFs and $\gamma \in \mathbb{F}$. Also, $f = \widehat{T}_1 + \cdots + \widehat{T}_s + \gamma$, where for all $k \in [s]$, $\widehat{T}_k = T_k(B\mathbf{x} + \mathbf{d})$. $T_1, \ldots, T_{s_1}$ are the good terms of $\mathtt{C}$, while, $\widehat{T}_1, \ldots, \widehat{T}_{s_1}$ are the good terms of f. Similarly, $T_{s_1+1}, \ldots, T_{s_2}$ are the bad terms of $\mathtt{C}$, while $\widehat{T}_{s_1+1}, \ldots, \widehat{T}_{s_2}$ are the bad terms of f. If $\mathtt{C}$ has a top dangling variable, it is $T_s = x_n$, $s' := s - 1$, and $T_{s_2+1} + \cdots + T_{s-1}$ is the top quadratic form. Otherwise, $T_{s_2+1} + \cdots + T_s$ is the top quadratic form and $s' := s$. If $\mathtt{C}$ has a top dangling variable, then $\ell := B \circ x_n + d_n$, where d_n is the n-th coordinate of $\mathbf{d}$.

It follows from Lemma 4.3 that after Step 3 is executed, $\widehat{T}_1(A\mathbf{x} + \mathbf{b}), \ldots, \widehat{T}_{s'}(A\mathbf{x} + \mathbf{b})$, and hence $\widehat{T}_1(A\mathbf{x}), \ldots, \widehat{T}_{s'}(A\mathbf{x})$ are variable disjoint while $\sum_{k=s_2+1}^{s'} \widehat{T}_k(A\mathbf{x}) = (y_1 + c_1)(y_2 + c_2) + \cdots + (y_{2m-1} + c_{2m-1})(y_{2m} + c_{2m})$, where $c_1, \ldots, c_{2m} \in \mathbb{F}$. Then, Claim 4.13 implies that after Step 11, $\mathbf{z}_1, \ldots, \mathbf{z}_{s_2}$ are variable sets of $\widehat{T}_1(A\mathbf{x}), \ldots, \widehat{T}_{s_2}(A\mathbf{x})$, respectively. Further, if there is a dangling variable, then Lemma 4.4 implies that $\ell(A\mathbf{x}) = u_0 + c$ for some $c \in \mathbb{F}$. However, now $f \in \mathrm{orb}(\mathtt{C}')$, where $\mathtt{C}'$ is as defined in the proof of Lemma 4.4. If $\mathtt{C}$ does not have a top-dangling variable, then let $\mathtt{C}' = \mathtt{C}$. We first show that after Step 26 is executed, $f(A\mathbf{x}) \in \text{PS-orb}(\mathtt{C}')$. As there are no redundant variables in $\widehat{T}_1(A\mathbf{x}), \ldots, \widehat{T}_{s'}(A\mathbf{x})$, for all $k \in [s_2]$, $|\mathbf{z}_k| = \left|\mathrm{var}\left(T'_k\right)\right|$ and $|\mathbf{y}| = \left|\mathrm{var}\left(\sum_{k=s_2+1}^{s} T'_k\right)\right|$. So there exists a permutation matrix $P_0 \in M(n, \mathbb{F})$ (that maps u_0 to u_0) such that for all $k \in [s_2]$, $\mathrm{var}(T'_k(P_0\mathbf{x})) = \mathbf{z}_k$ and $\mathrm{var}\left(\sum_{k=s_2+1}^{s} T'_k(P_0\mathbf{x})\right) = \mathbf{y}$. There exists a $B' \in \mathrm{GL}(n, \mathbb{F})$ and $\mathbf{d}' \in \mathbb{F}^n$ such that $f(A\mathbf{x}) = \mathtt{C}'(P_0(B'\mathbf{x} + \mathbf{d}'))$. Notice that it suffices to prove that $f(A\mathbf{x}) \in \text{PS-orb}(\mathtt{C}'(P_0\mathbf{x}))$. We now analyse the for loop of lines 15-25. For any $k \in [s_2]$, as the k-th iteration of the loop only works on $\widehat{T}_k(A\mathbf{x})$ and $\mathbf{z}_k$, we can look at it in isolation.

Claim 4.18. *For any $k \in [s_2]$, after the execution of the k-th iteration of the for loop of lines 15-25 there exists a permutation matrix $P_k \in M(|\mathbf{z}_k|, \mathbb{F})$, an invertible scaling matrix $S_k \in M(|\mathbf{z}_k|, \mathbb{F})$, and a $\mathbf{b}_k \in \mathbb{F}^{|\mathbf{z}_k|}$ such that $\widehat{T}_k\left(A\left(A_k\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k\right)\right) = T'_k(P_0\left(P_kS_k\mathbf{z}_k + \mathbf{b}_k, \mathbf{x} \setminus \mathbf{z}_k\right))$.*

The proof of this claim is given in Section D.18; here we finish the proof of the lemma assuming the claim. After Step 14, there already exists a permutation matrix $P_{s_2+1} \in M(|\mathbf{y}|, \mathbb{F})$, an invertible scaling matrix $S_{s_2+1} \in M(|\mathbf{y}|, \mathbb{F})$ (such that $P_{s_2+1}S_{s_2+1} \circ u_0 = u_0$) and a $\mathbf{b}_{s_2+1} \in \mathbb{F}^{|\mathbf{y}|}$ such that

$$\sum_{k=s_2+1}^{s} \widehat{T}_k\left(A\mathbf{x}\right) = \sum_{k=s_2+1}^{s} T'_k(P_0\left(P_{s_2+1}S_{s_2+1}\mathbf{y} + \mathbf{b}_{s_2+1}, \mathbf{z}\right)).$$

Let $P \in M(|\mathbf{z}|, \mathbb{F})$ be a permutation matrix that maps every $z \in \mathbf{z}_k$ to $P_k \circ z$ for all $k \in [s_2]$ and every $y \in \mathbf{y}$ to $P_{s_2+1} \circ y$. Similarly, let $S \in M(|\mathbf{z}|, \mathbb{F})$ be a scaling matrix that maps every $z \in \mathbf{z}_k$ to $S_k \circ z$ for all $k \in [s_2]$ and and every $y \in \mathbf{y}$ to $S_{s_2+1} \circ y$. Also, let $\mathbf{b} \in \mathbb{F}^n$ be such that for all $k \in [s_2]$, its coordinates corresponding to $\mathbf{z}_k$ are $\mathbf{b}_k$ and those corresponding to $\mathbf{y}$ are $\mathbf{b}_{s_2+1}$. As A'_0 maps every $z \in \mathbf{z}_k$ to $A_k \circ z$, $\forall k \in [s_2]$ and maps every $y \in \mathbf{y}$ to itself, after A is set to AA' we have that $\widehat{T}_k(A\mathbf{x}) = T'(P_0(PS\mathbf{x} + \mathbf{b}))$ yielding $f(A\mathbf{x}) = \mathtt{C}'(P_0\left(PS\mathbf{x} + \mathbf{b}\right)) \in \text{PS-orb}\left(\mathtt{C}'\right)$.

If $\mathtt{C}' = \mathtt{C}$, then we are done. Otherwise, in Step 29 $f(A\mathbf{x})$ is reconstructed. From Lemma F.1, we have that the terms of the reconstructed ROF f' are constant multiples of $T'_1(P_0\left(PS\mathbf{x} + \mathbf{b}\right)), \ldots, T'_s(P_0\left(PS\mathbf{x} + \mathbf{b}\right))$. In fact, as $T'_1(P_0\left(PS\mathbf{x} + \mathbf{b}\right)), \ldots, T'_s(P_0\left(PS\mathbf{x} + \mathbf{b}\right))$ are variable disjoint and hence linearly independent, the terms are exactly $T'_1(P_0\left(PS\mathbf{x} + \mathbf{b}\right)), \ldots, T'_s(P_0\left(PS\mathbf{x} + \mathbf{b}\right))$. Fix any k such that $T_k \neq T'_k$. Recall that in this case, $T_k = zQ_{k,2} \cdots Q_{k,m_k}$ and $T'_k = z\left(Q_{k,2} \cdots Q_{k,m_k} + c_k\right)$. From Lemma F.1, the corresponding term of f' is $(c\alpha'_1 x + c\alpha'_0)\left(c^{-1}\left(Q_{k,2} \cdots Q_{k,m_k}\right)\left(P_0\left(PS\mathbf{x} + \mathbf{b}\right)\right) + c^{-1}c_k\right)$,

where $\alpha_1' x = P_0 PS \circ z$, α_0' is the z-th entry of $P_0\mathbf{b}$, and $c \neq 0$. Hence in Step 29, $\alpha_1 = c\alpha_1'$ and $\beta = c^{-1}c_k$. Notice that $P_0 PS \circ u_0 = u_0$. So after A is updated to map u_0 to $u_0 - \alpha_1\beta x$, $\widehat{T}_k(A\mathbf{x}) = T_k(P_0(PS\mathbf{x} + \mathbf{b}))$ yielding $f(A\mathbf{x}) \in \text{PS-orb}(\mathtt{C})$. □

4.3.6 Running time of Algorithm 1

Notice that whenever a recursive call is made to Find-Equivalence(), it is for a polynomial in the orbit of a distinct $+$-rooted sub-ROF or variable of the original ROF $\mathtt{C}$. As there are at most n many $+$-rooted sub-ROFs and n variables, there are at most $2n$ many recursive calls. Thus to prove that Find-Equivalence() runs $\text{poly}(n)$ time we only need to argue that the time required by each recursive call (not counting the time spent in any sub-calls) is $\text{poly}(n)$. We divide this time into three parts: the time required to query the black-box of the input polynomial, time required to prepare black-boxes for sub-calls, and the time required to do everything else. The last of these is $\text{poly}(n)$ because the Procedures 2, 3, 6, 5, and Algorithm 13 run in time $\text{poly}(n)$. This is so as all the operations that they perform like sparse polynomial interpolation, computing partial derivatives of order at most two, computing determinants of symbolic matrices, and factoring polynomials can be done efficiently in black-box fashion.

We now analyze how much time is required to query the black-box of the input polynomial and prepare black-boxes for sub-calls. To do this, let us understand how the black-boxes for the factors of the terms $\widehat{T}_1(A\mathbf{x}), \ldots, \widehat{T}_{s_2}(A\mathbf{x})$ are prepared in the for loop of lines 15-25 in first call to Find-Equivalence(), i.e., the call for f. Observe that for any $k \in [s_2]$, Compute-Term-Black-Box() obtains black-box access to $\widehat{T}_k(A\mathbf{x})$ by setting all variables other than those in $\mathbf{z}_k$ to 0 in $f(A\mathbf{x})$ and subtracting a known constant β from the resulting polynomial. Thus black-box access to $\widehat{T}_k(A\mathbf{x})$ is obtained by evaluating f at known affine forms $\ell_1, \ldots, \ell_n$ (obtained from A by setting variables not in $\mathbf{z}_k$ to 0) and subtracting a known constant β from $f(\ell_1, \ldots, \ell_n)$.

For any $l \in [m_k]$, to obtain black-box access to the factors of $\widehat{T}_k(A\mathbf{x})$, we first compute a matrix $A_{k,0} \in \text{GL}(|\mathbf{z}_k|, \mathbb{F})$ such that the factors $\widehat{Q}_1, \ldots, \widehat{Q}_{m_k}$ of $\widehat{T}_k(A(A_{k,0}\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k))$ are variable disjoint. To obtain black-box access to $\widehat{Q}_l$ for some $l \in [m_k]$, we first set the variables in $\mathbf{z}_k \setminus \mathbf{z}_{k,l}$ to random field elements $\mathbf{a}'$ and compute the constant β_l from the (possibly inefficient) black-boxes of $\widehat{Q}_1, \ldots, \widehat{Q}_{m_k}$ obtained from $\widehat{T}_k(A(A_{k,0}\mathbf{z}, \mathbf{x} \setminus \mathbf{z}))$ using the black-box factorisation algorithm in [KT90]. We then compute $\beta_l^{-1} \cdot \widehat{T}\left(A_{k,0}\left(\mathbf{z}_{k,l}, \mathbf{z}_k \setminus \mathbf{z}_{k,l} = \mathbf{a}'\right)\right)$. Notice that this is the same as evaluating f at known affine forms $\ell_1', \ldots, \ell_n'$ (obtained from $\ell_1, \ldots, \ell_n$ by setting $\mathbf{z}_k \setminus \mathbf{z}_{k,l} = \mathbf{a}'$), multiplying it by a known constant β_l^{-1} and subtracting a known constant $\beta_l^{-1}\beta$ from it.

In any recursive call to Find-Equivalence(), the black-boxes for sub-calls are prepared in the same way. Thus the discussion in the above paragraph implies that no matter the recursive depth for a recursive call for a polynomial f', the black-box for f' would look like $\alpha f(\ell_1, \ldots, \ell_n) - \beta$, where α, β are known constants and $\ell_1, \ldots, \ell_n$ known affine forms in $\text{var}(f')$. Thus the time to query the black-box of f' is $\text{poly}(n)$; not $\text{poly}(|\text{var}(f')|)$, but still independent of the recursive depth. Similarly the time required to prepare black-boxes for sub-calls is also $\text{poly}(n)$ and independent of the recursion depth as all that needs to be done is to compute appropriate affine forms $\ell_1', \ldots, \ell_n'$ and constants α' and β'. Thus the algorithm runs in time $\text{poly}(n)$.

5 Polynomial equivalence for orbits of ROFs

In this section, we shall prove Theorem 2. Let ROF_0 be the class of all additive-constant-free canonical ROFs.

5.1 The algorithm

The following algorithm decides whether $f_1(\mathbf{x}), f_2(\mathbf{x}) \in \text{orb}(\text{ROF}_0)$ are equivalent or not.

Algorithm 7 Equivalence-Test($f_1(\mathbf{x}), f_2(\mathbf{x})$)

Input: Black-box access to $f_1(\mathbf{x}), f_2(\mathbf{x}) \in \text{orb}(\text{ROF}_0)$.
Output: Whether or not f_1 and f_2 are equivalent. If they are equivalent, then $A \in \text{GL}(n, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^n$ such that $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$.

/* Reconstructing canonical ROFs equivalent to f_1 and f_2. */

1. **for** $i \in [2]$ **do**
2. $A_i \leftarrow$ Find-Equivalence($f_i(\mathbf{x})$) (Algorithm 1).
3. $\mathtt{C}_i' \leftarrow$ Reconstruct-ROF($f_i(A_i\mathbf{x})$) (Algorithm 13).
4. $S_i, \mathbf{b}_i \leftarrow$ Canonize($\mathtt{C}_i'$) (Procedure 14), where $S_i \in \text{GL}(n, \mathbb{F})$ is a scaling matrix, $\mathbf{b}_i \in \mathbb{F}^n$.
5. $\mathtt{C}_i \leftarrow \mathtt{C}_i'(S_i\mathbf{x} + \mathbf{b}_i)$, $G_i \leftarrow$ the underlying tree of $\mathtt{C}_i$ wherein all internal nodes are unlabelled and the leaves are labelled by variables.
6. **end for**

/* Checking if $\mathtt{C}_1$ and $\mathtt{C}_2$ are equivalent */

7. **if** G_1 and G_2 are isomorphic as rooted trees **then**
8. If σ is the permutation such that $\sigma(G_2) = G_1$, construct a permutation matrix P that maps x_i to $\sigma(x_i)$ $\forall i \in [n]$ using Fact A.4. $A \leftarrow A_2S_2PS_1^{-1}A_1^{-1}$, $\mathbf{b} \leftarrow A_2\mathbf{b}_2 - A_2S_2PS_1^{-1}\mathbf{b}_1$.
9. Use the Schwartz-Zippel Lemma to check if $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$. If yes, return EQUIVALENT, A and $\mathbf{b}$. Else, return NOT EQUIVALENT.
10. **else**
11. Return NOT EQUIVALENT.
12. **end if**

5.2 Analysis of the algorithm

We establish the correctness of the above algorithm by proving the following lemma.

Lemma 5.1 (Correctness of Algorithm 7)**.** *Given black-box access to two n-variate polynomials $f_1(\mathbf{x})$, $f_2(\mathbf{x}) \in \text{orb}(ROF_0)$, Algorithm 7 correctly determines with high probability whether they are equivalent or not provided that* $\text{char}(\mathbb{F}) = 0$ *or* $\geq n^2$ *and* $|\mathbb{F}| \geq n^{13}$*. Moreover, if they are equivalent, it returns an $A \in \text{GL}(n, \mathbb{F})$ and a $\mathbf{b} \in \mathbb{F}^n$ such that $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$.*

Proof. If $f_1 \notin \text{orb}(f_2)$, then Step 9 ensures that the algorithm returns NOT EQUIVALENT with high probability. So suppose that $f_1 \in \text{orb}(f_2)$. In this case, there exists a $\mathtt{C} \in \text{ROF}_0$ such that $f_1, f_2 \in \text{orb}(\mathtt{C})$. Then, $f_1(A_1\mathbf{x}), f_2(A_2\mathbf{x}) \in \text{PS-orb}(\mathtt{C})$ (from Lemma 4.1), and so the only non-zero

additive-constants in them are translations, i.e. constants attached to $+$ gates which have a variable as a child. As, from Lemma F.1, $\mathtt{C}_1'$ and $f_1(A_1\mathbf{x})$, $\mathtt{C}_2'$ and $f_2(A_2\mathbf{x})$ are equal up to scaling of the leaves, the only non-zero additive-constants in $\mathtt{C}_1'$ and $\mathtt{C}_2'$ are also translations. We now use this fact to show that $\mathtt{C}_1$ and $\mathtt{C}_2$ are the same as $\mathtt{C}$ up to a permutation of variables.

As mentioned above, $f_1(A_1\mathbf{x}), f_2(A_2\mathbf{x}) \in$ PS-orb$(\mathtt{C})$ and $\mathtt{C}_1', \mathtt{C}_2'$ are same as $f_1(A_1\mathbf{x}), f_2(A_2\mathbf{x})$ up to scaling of leaves. This means that $\mathtt{C}_1', \mathtt{C}_2'$ can be obtained from $\mathtt{C}$ by permuting, scaling, and translating the variables and scaling the additive-constants (as they are also leaves of $\mathtt{C}_1', \mathtt{C}_2'$). However as the only non-zero additive constants in $\mathtt{C}_1'$ and $\mathtt{C}_2'$ are translations, these two ROFs differ from $\mathtt{C}$ by just permutation, scaling and translation of the variables. From Observation F.1, $\mathtt{C}_1$ and $\mathtt{C}_2$ are constant-free regular ROFs obtained from $\mathtt{C}_1'$ and $\mathtt{C}_2'$ by recovering the scaling and translation of variables. Hence, they must be equal to $\mathtt{C}$ up to a permutation of variables.

As $\mathtt{C}_1$ and $\mathtt{C}_2$ are the same up to a permutation of variables, their underlying trees G_1 and G_2 are isomorphic as rooted trees. So, a permutation σ such that $\sigma(G_2) = G_1$ exists. As σ is completely determined by its restriction to the leaves of G_2, if P is as defined in Step 8, then $\mathtt{C}_1(\mathbf{x}) = \mathtt{C}_2(P\mathbf{x})$. A simple calculation shows that this implies $f_1(\mathbf{x}) = f_2(A\mathbf{x}+\mathbf{b})$ for A and $\mathbf{b}$ defined in Step 8. □

Running time of the algorithm. Find-Equivalence(), Reconstruct-ROF(), and Canonize() run in time polynomial in n. Also as mentioned in Fact A.4, a polynomial time algorithm exists for the rooted tree isomorphism problem. Moreover, the Schwartz-Zippel lemma also yields a polynomial time algorithm for checking if $f_1(\mathbf{x}) = f_2(A\mathbf{x}+\mathbf{b})$ in Step 9. Thus, Algorithm 7 runs in time poly(n). This along with Lemma 5.1 proves Theorem 2.

6 Conclusion

In this work, we give the first randomized polynomial-time equivalence test for ROFs (Theorem 1) and use this result to solve PE for orbits of (slightly restricted) ROFs (Theorem 2). These results are substantial generalizations of two well-studied problems in algebraic complexity, namely quadratic form equivalence and reconstruction of ROFs. As PE is graph isomorphism hard for even cubic forms, it is indeed satisfying to know that PE can be solved efficiently for an unbounded-depth, unbounded-degree, and unbounded-fanin circuit class such as orbits of ROFs. Theorem 1 also implies efficient learning of random arithmetic formulas (without any restriction on the underlying tree structure) in the high number of variables setting.

The algorithms are based on a novel interplay between a few crucial properties of the factors and the essential variables of the Hessian determinant of an ROF, the essential variables of the ROF, and certain structures in the ROF called "skewed paths". Proving these properties of the Hessian and combining them effectively with the skewed paths to make up for the dearth of essential variables and to recursively discover formulas in the orbits of sub-ROFs of lower depth (without blowing up the complexity exponentially due to unbounded depth) constitute the main technical contributions of this work. The approach developed in this work and the insights obtained thus may turn out to be independently useful for learning other related circuit models.

We end this section by noting a few future directions that can be pursued:

1. **Generalizing our results.** We believe that the mild "additive-constant-free" restriction on the ROFs in Theorems 2 can be removed completely by building on the techniques of this work. Indeed, we show in Section E that this relaxation is possible for depth-4 ROFs. It

is worth showing the same for general ROFs. Another interesting generalization of Theorem 1 would be an equivalence test for power-substituted ROFs and, more generally, for *univariate-substituted* ROFs [35]. An equivalence test for univariate-substituted ROFs would greatly generalize the equivalence test for the sums of univariates model studied in [GKP18] and the reconstruction algorithm for preprocessed ROFs in [SV14]. We believe that our equivalence test and its analysis can be extended to work for univariate-substituted ROFs. To support this belief, let us consider the power-substituted sum-product polynomial $\mathsf{SPP} := \sum_{i\in[s]} \prod_{j\in[d]} x_{i,j}^{e_{i,j}}$, where $e_{i,j} \in \mathbb{N}$. It turns out that $\det(H_{\mathsf{SPP}})$ factorizes as:

$$\det(H_{\mathsf{SPP}}) = (-1)^{s(d-1)} \cdot \prod_{i\in[s],j\in[d]} e_{i,j} \cdot \prod_{i\in[s]} (e_{i,1} + \ldots + e_{i,d} - 1) \cdot \prod_{i\in[s],j\in[d]} x_{i,j}^{e_{i,j}\cdot d-2}.$$

So the equivalence test for SP, described in Section 1.3.1, works (almost as it is) for SPP.

2. **Learning orbits of sparse polynomials and ROABPs.** As mentioned in Section 1.1.2, studying the orbit of a circuit class is a natural first step towards understanding affine projections of the class. Efficient proper learning algorithms are long known for sparse polynomials [KS01] and ROABPs [BBB+00, KS06]. Recall that affine projections of these classes capture immensely powerful circuit classes such as depth-3 circuits and ABPs. Like ROFs, can we design efficient learning algorithms for *orbits* of sparse polynomials and ROABPs?

3. **Learning random formulas.** Theorem 1 solves the learning problem for random formulas when the number of variables n is larger than the size s of the underlying tree of the formula. A more interesting setting of parameters is $s = \text{poly}(n)$. Can we design an efficient learning algorithm for random formulas (of even constant depth) if $s \gg n$?

References

[AHK93] Dana Angluin, Lisa Hellerstein, and Marek Karpinski. Learning Read-Once Formulas with Queries. *J. ACM*, 40(1):185–210, 1993. 13

[AHU83] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *Data Structures and Algorithms*. Addison-Wesley, 1983. 11, 50

[Ara11] Manuel Araújo. Classification of quadratic forms. https://www.math.tecnico.ulisboa.pt/~ggranja/manuel.pdf, 2011. 50

[AS05] Manindra Agrawal and Nitin Saxena. Automorphisms of finite rings and applications to complexity of problems. In *23rd Annual Symposium on Theoretical Aspects of Computer Science, STACS 2005*, pages 1–17, 2005. 12

[AS06] Manindra Agrawal and Nitin Saxena. Equivalence of f-algebras and cubic forms. In *23rd Annual Symposium on Theoretical Aspects of Computer Science, STACS 2006*, pages 115–126, 2006. 12

[35] A univariate-substituted ROF is obtained from an ROF by substituting every variable x_i by an arbitrary (and unknown) univariate polynomial $g_i(x_i)$. Such ROFs were called *preprocessed* ROFs in [SV14].

[BB98] Daoud Bshouty and Nader H. Bshouty. On Interpolating Arithmetic Read-Once Formulas with Exponentiation. *J. Comput. Syst. Sci.*, 56(1):112–124, 1998. Conference version appeared in the proceedings of COLT 1994. 13

[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. Conference version appeared in the proceedings of FOCS 1996. 43

[BC98] Nader H. Bshouty and Richard Cleve. Interpolating Arithmetic Read-Once Formulas in Parallel. *SIAM J. Comput.*, 27(2):401–413, 1998. Conference version appeared in the proceedings of FOCS 1992. 13

[Ber70] Elwyn R Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24:713–735, 1970. 49

[BFP15] Jérémy Berthomieu, Jean-Charles Faugère, and Ludovic Perret. Polynomial-time algorithms for quadratic isomorphism of polynomials: The regular case. *J. Complex.*, 31(4):590–616, 2015. 12

[BG21] Vishwas Bhargava and Sumanta Ghosh. Improved Hitting Set for Orbit of ROABPs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 30:1–30:23, 2021. 2

[BGKS21] Vishwas Bhargava, Ankit Garg, Neeraj Kayal, and Chandan Saha. Learning generalized depth-three arithmetic circuits in the non-degenerate case. *Electron. Colloquium Comput. Complex.*, page 155, 2021. 13

[BHH95a] Nader H. Bshouty, Thomas R. Hancock, and Lisa Hellerstein. Learning arithmetic read-once formulas. *SIAM J. Comput.*, 24(4):706–735, 1995. Conference version appeared in the proceedings of STOC 1992. 3, 4, 13, 79

[BHH95b] Nader H. Bshouty, Thomas R. Hancock, and Lisa Hellerstein. Learning Boolean Read-Once Formulas over Generalized Bases. *J. Comput. Syst. Sci.*, 50(3):521–542, 1995. Conference version appeared in the proceedings of COLT 1992. 13

[BRS17] Markus Bläser, B. V. Raghavendra Rao, and Jayalal Sarma. Testing Polynomial Equivalence by Scaling Matrices. In *Proceedings of 21st International Symposium on Fundamentals of Computation Theory (FCT), France*, volume 10472, pages 111–122, 2017. 13

[BW15] Peter A Brooksbank and James B Wilson. The module isomorphism problem reconsidered. *Journal of Algebra*, 421:541–559, 2015. 12

[Car06] Enrico Carlini. Reducing the number of variables of a polynomial. In *Algebraic Geometry and Geometric Modeling*, pages 237–247, 2006. 14, 17

[CCL10] Jin-yi Cai, Xi Chen, and Dong Li. Quadratic Lower Bound for Permanent Vs. Determinant in any Characteristic. *Comput. Complex.*, 19(1):37–56, 2010. Conference version appeared in the proceedings of STOC 2008. 17

[CMM17] Sunil K. Chebolu, Dan McQuillan, and Ján Mináč. Witt's cancellation theorem seen as a cancellation. *Expositiones Mathematicae*, 35(3):300–314, 2017. 1

[DdOS14] Zeev Dvir, Rafael Mendes de Oliveira, and Amir Shpilka. Testing Equivalence of Polynomials under Shifts. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 417–428. Springer, 2014. 13

[FGS18] Vyacheslav Futorny, Joshua Grochow, and Vladimir Sergeichuk. Wildness for tensors. *Linear Algebra and its Applications*, 566, 12 2018. 12

[GGKS19] Ankit Garg, Nikhil Gupta, Neeraj Kayal, and Chandan Saha. Determinant equivalence test over finite fields and over Q. In *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Greece*, volume 132 of *LIPIcs*, pages 62:1–62:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 12, 13

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic Circuits: A Chasm at Depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. Conference version appeared in the proceedings of FOCS 2013. 2

[GKL11] Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Efficient reconstruction of random multilinear formulas. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 778–787. IEEE Computer Society, 2011. 3

[GKP18] Ignacio García-Marco, Pascal Koiran, and Timothée Pecatte. Polynomial Equivalence Problems for Sum of Affine Powers. In *Proceedings of the 2018 ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC 2018, New York, NY, USA, July 16-19, 2018*, pages 303–310. ACM, 2018. 13, 17, 43

[GKQ14] Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random arithmetic formulas can be reconstructed efficiently. *Comput. Complex.*, 23(2):207–303, 2014. Conference version appeared in the proceedings of CCC 2013. 3, 12

[GKS20] Ankit Garg, Neeraj Kayal, and Chandan Saha. Learning sums of powers of low-degree polynomials in the non-degenerate case. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 889–899. IEEE, 2020. 2, 3, 12

[GQ21] Joshua A. Grochow and Youming Qiao. On the complexity of isomorphism problems for tensors, groups, and polynomials I: tensor isomorphism-completeness. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 31:1–31:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 12

[GQT21] Joshua A. Grochow, Youming Qiao, and Gang Tang. Average-case algorithms for testing isomorphism of polynomials, algebras, and multilinear forms. In *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 38:1–38:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 12

[Gro12] Joshua Abraham Grochow. *Symmetry and equivalence relations in classical and geometric complexity theory*. PhD thesis, Department of Computer Science, The University of Chicago, Chicago, Illinois, 2012. 12, 13

[GS19] Nikhil Gupta and Chandan Saha. On the symmetries of and equivalence test for design polynomials. In *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 53:1–53:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. 12, 13

[Gur77] V. A. Gurvich. On repetition-free boolean functions. *Uspekhi Matematicheskikh Nauk*, 32(1):183–184, 1977. (in Russian). 13

[HH91] Thomas R. Hancock and Lisa Hellerstein. Learning read-once formulas over fields and extended bases. In Manfred K. Warmuth and Leslie G. Valiant, editors, *Proceedings of the Fourth Annual Workshop on Computational Learning Theory, COLT 1991, Santa Cruz, California, USA, August 5-7, 1991*, pages 326–336. Morgan Kaufmann, 1991. 3, 4, 13, 79

[HHTT22] Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. Depth-d Threshold Circuits vs. Depth-$(d+1)$ AND-OR Trees. *Electronic Colloquium on Computational Complexity (ECCC)*, page 87, 2022. 13

[HRST17] Johan Håstad, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An Average-Case Depth Hierarchy Theorem for Boolean Circuits. *J. ACM*, 64(5):35:1–35:27, 2017. 13

[IQ19] Gábor Ivanyos and Youming Qiao. Algorithms Based on *-Algebras, and Their Applications to Isomorphism of Polynomials with One Secret, Group Isomorphism, and Polynomial Identity Testing. *SIAM J. Comput.*, 48(3):926–963, 2019. Conference version appeared in the proceedings of SODA 2018. 1, 12

[JQSY19] Zhengfeng Ji, Youming Qiao, Fang Song, and Aaram Yun. General linear group action on tensors: A candidate for post-quantum cryptography. In *Theory of Cryptography - 17th International Conference, TCC 2019, Nuremberg, Germany, December 1-5, 2019, Proceedings, Part I*, volume 11891 of *Lecture Notes in Computer Science*, pages 251–281. Springer, 2019. 1

[Kay11] Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1409–1421. SIAM, 2011. 5, 6, 13, 14, 17

[Kay12] Neeraj Kayal. Affine projections of polynomials: extended abstract. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 643–662, 2012. 12, 13

[KLN+93] Mauricio Karchmer, Nathan Linial, Ilan Newman, Michael E. Saks, and Avi Wigderson. Combinatorial characterization of read-once formulae. *Discret. Math.*, 114(1-3):275–282, 1993. 13

[KNS19] Neeraj Kayal, Vineet Nair, and Chandan Saha. Average-case linear matrix factorization and reconstruction of low width algebraic branching programs. *Comput. Complex.*, 28(4):749–828, 2019. 3, 13, 64

[KNST19] Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of full rank algebraic branching programs. *ACM Trans. Comput. Theory*, 11(1):2:1–2:56, 2019. Conference version appeared in the proceedings of CCC 2017. 12, 13, 14, 17, 49

[KS01] Adam R. Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 216–223, 2001. 5, 8, 21, 22, 43

[KS06] Adam R. Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of Computing*, 2(10):185–206, 2006. Conference version appeared in the proceedings of COLT 2003. 43

[KS19] Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 413–424. ACM, 2019. 3, 12

[KS21a] Pascal Koiran and Subhayan Saha. Black Box Absolute Reconstruction for Sums of Powers of Linear Forms. *CoRR*, abs/2110.05305, 2021. 13

[KS21b] Pascal Koiran and Mateusz Skomra. Derandomization and absolute reconstruction for sums of powers of linear forms. *Theor. Comput. Sci.*, 887:63–84, 2021. 13

[KT90] Erich Kaltofen and Barry M. Trager. Computing with Polynomials Given By Black Boxes for Their Evaluations: Greatest Common Divisors, Factorization, Separation of Numerators and Denominators. *J. Symb. Comput.*, 9(3):301–320, 1990. Conference version appeared in the proceedings of FOCS 1988. 4, 5, 10, 11, 19, 25, 38, 40, 49, 82

[Lam04] T. Y. Lam. *Introduction To Quadratic Forms Over Fields*. American Mathematical Society, 2004. 50

[LLL82] Arjen K Lenstra, Hendrik W Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. 49

[MNS20] Janaky Murthy, Vineet Nair, and Chandan Saha. Randomized Polynomial-Time Equivalence Between Determinant and Trace-IMM Equivalence Tests. In *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 72:1–72:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 12, 13

[MR04] Thierry Mignon and Nicolas Ressayre. A quadratic bound for the determinant and permanent problem. *International Mathematics Research Notes*, 2004(79):4241–4253, 2004. 17

[MS21] Dori Medini and Amir Shpilka. Hitting sets and reconstruction for dense orbits in VP_e and $\Sigma\Pi\Sigma$ circuits. In *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 19:1–19:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 2, 4, 12, 13

[MV18] Daniel Minahan and Ilya Volkovich. Complete derandomization of identity testing and reconstruction of read-once formulas. *ACM Trans. Comput. Theory*, 10(3):10:1–10:11, 2018. Conference version appeared in the proceedings of CCC 2017. 3, 4, 13, 79

[Pat96] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In *Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding*, pages 33–48, 1996. 1, 12

[RS11] B. V. Raghavendra Rao and Jayalal Sarma. Isomorphism testing of read-once functions and polynomials. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2011, December 12-14, 2011, India*, volume 13 of *LIPIcs*, pages 115–126. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011. 13

[Sax06] Nitin Saxena. *Morphisms of rings and applications to complexity*. PhD thesis, Indian Institute of Technology, Kanpur, 2006. 1

[Sch80] Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM*, 27(4):701–717, 1980. 11

[Ser73] Jean-Pierre Serre. *A course in arithmetic*. Springer, 1973. 50

[Sip83] Michael Sipser. Borel Sets and Circuit Complexity. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston, Massachusetts, USA*, pages 61–69. ACM, 1983. 13

[ST21] Chandan Saha and Bhargav Thankey. Hitting Sets for Orbits of Circuit Classes and Polynomial Families. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 50:1–50:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 2, 13

[SV14] Amir Shpilka and Ilya Volkovich. On Reconstruction and Testing of Read-Once Formulas. *Theory of Computing*, 10(18):465–514, 2014. Conference version appeared in the proceedings of STOC 2008. 3, 4, 13, 43, 79

[Tav15] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. Conference version appeared in the proceedings of MFCS 2013. 2

[Thi98] Thomas Thierauf. The isomorphism problem for read-once branching programs and arithmetic circuits. *Chicago J. Theor. Comput. Sci.*, 1998, 1998. 1

[Vol16] Ilya Volkovich. Characterizing arithmetic read-once formulae. *ACM Trans. Comput. Theory*, 8(1):2:1–2:19, 2016. 3, 13

[Wal13] Lars Ambrosius Wallenborn. Computing the hilbert symbol, quadratic form equivalence and integer factoring. Diploma thesis, Rheinischen Friedrich-Wilhelms-Universität Bonn, 2013. 50

[Wit37] Ernst Witt. Theorie der quadratischen Formen in beliebigen Körpern. *J. Reine Angew. Math.*, 176:31–44, 1937. 1

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, pages 216–226, 1979. 11

Notations	Usage
$\mathtt{C}$, Δ	An ROF and its product depth, respectively
T, Q (with or without subscripts)	$\times$-rooted and +-rooted sub-ROFs, respectively
A, B, C, P, R, S	Matrices over $\mathbb{F}$
U, W	Spaces spanned by the first order partials of polynomials
E, F, I, J, N, V	Sets
f, g, h, p, q, ℓ, r	Polynomials
t, u, x, y, z	Variables
$\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{u}$	Sets of variables
α, β, γ, c	Elements of $\mathbb{F}$
$d, e, i, j, k, l, m, n, s$	Natural numbers
$\mathbf{a}, \mathbf{b}, \mathbf{d}, \boldsymbol{\alpha}$	Vectors over $\mathbb{F}$

Table 1: Notations

A Some useful algorithmic facts

A.1 Black-box polynomial factorization

Fact A.1 (Black-box for partials)**.** *Let $d \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > d$, and $g \in \mathbb{F}[\mathbf{x}]$ be a degree d polynomial given as a black-box. Then, for $x \in \mathbf{x}$, a black-box for $\frac{\partial g}{\partial x}$ can be computed in $\mathrm{poly}(|\mathbf{x}|, d)$ time.*

The above fact is well-known; a proof of it can be found in Section 2.2 of [KNST19].

Fact A.2 (Black-box polynomial factorization [KT90])**.** *Let $d \in \mathbb{N}$, $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > d$, and $|\mathbb{F}| \geq d^6$. There is a randomized algorithm, with oracle access to univariate polynomial factorization over $\mathbb{F}$, that takes input black-box access to a polynomial $g \in \mathbb{F}[\mathbf{x}]$ of degree d and outputs black-boxes for the irreducible factors of g in $\mathrm{poly}(|\mathbf{x}|, d)$ time.*

Remark. Since our model of computation allows univariate polynomial factorization, we will assume that black-box polynomial factorization can be done in randomized polynomial-time. This assumption is justified particularly for finite fields and $\mathbb{Q}$ [Ber70, LLL82].

A.2 Quadratic form equivalence

Known algorithms for quadratic form equivalence (QFE) over $\mathbb{C}, \mathbb{R}, \mathbb{Q}$, and finite fields are based on well-known classification of quadratic forms. Refer to [Ser73, Lam04, Ara11] for a comprehensive discussion on this. We record the complexity of QFE over these fields in the fact below. Over $\mathbb{R}$ and $\mathbb{C}$, the model of computation is an arithmetic circuit with oracle access to a square root finding algorithm; every operation in the circuit takes a unit time. Whereas, over $\mathbb{Q}$ and finite fields, the model of computation is a Turing machine, i.e., the running time is measured as bit operations.

Fact A.3 (Complexity of QFE)**.** *Let n be the number of variables in each of the two input quadratic forms.*

1. (Over $\mathbb{C}$ and $\mathbb{R}$). *There is a deterministic* $\mathrm{poly}(n)$ *time QFE algorithm.*
2. (Over finite fields). *Let* $\mathrm{char}(\mathbb{F}) \neq 2$. *There is a randomized* $\mathrm{poly}(n, \log|\mathbb{F}|)$ *time QFE algorithm.*
3. (Over $\mathbb{Q}$) [Wal13]. *There is a deterministic* $\mathrm{poly}(n, b)$ *time QFE algorithm with oracle access to integer factoring, where b is the bit length of the coefficients of the input quadratic forms.*

A.3 Rooted tree isomorphism

Definition A.1 (Tree isomorphism)**.** Two rooted trees $G_1 = (V_1, E_1, v_1)$ and $G_2 = (V_2, E_2, v_2)$ are *isomorphic*, if there is a bijection $\pi : V_1 \to V_2$ s.t. $(v, v') \in E_1 \Leftrightarrow (\pi(v), \pi(v')) \in E_2$ and $\pi(v_1) = v_2$.

Fact A.4 (Efficient tree isomorphism [AHU83])**.** *There is an algorithm that takes input two rooted trees $G_1 = (V_1, E_1, v_1)$ and $G_2 = (V_2, E_2, v_2)$ and decides if G_1 and G_2 are isomorphic. If the answer is yes, it also outputs an isomorphism. The running time of the algorithm is* $\mathrm{poly}(|V_1|, |V_2|)$.

The following observation is easy to show.

Observation A.1. *Let $G_1 = (V_1, E_1, v_1)$ and $G_2 = (V_2, E_2, v_2)$ be rooted trees and π an isomorphism from G_1 to G_2 s.t. $\pi(v_1) = v_2$. Then, π is completely determined by its restriction to the leaves of G_1.*

B Missing proofs from Section 2

B.1 Proof of Observation 2.1

Let x be a truly essential variable of g. Suppose there exists $\alpha_{x'} \in \mathbb{F}$, for every $x' \in \mathbf{x}$, such that $\sum_{x' \in \mathbf{x} \setminus \{x\}} \alpha_{x'} \frac{\partial g}{\partial x'} = \alpha_x \frac{\partial g}{\partial x}$, where $\alpha_x \neq 0$. Then, it follows from Fact 2.1 that there is an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ which maps x to itself and $g(A\mathbf{x})$ is x-free. Hence, by Definition 2.2, x is not truly essential.

Suppose x is not a truly essential variable of g. Then, there exists an $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ that maps x to itself and $g(A\mathbf{x})$ is x-free. Suppose the column of A labelled by x is $(\alpha_{y,x})^T_{y \in \mathbf{x}}$. Then, it follows from the chain rule of derivatives that $0 = \frac{\partial g(A\mathbf{x})}{\partial x} = \sum_{y \in \mathbf{x}} \alpha_{y,x} \frac{\partial g}{\partial y}(A\mathbf{x})$, which implies $\sum_{y \in \mathbf{x}} \alpha_{x,y} \frac{\partial g}{\partial y} = 0$. As A maps x to itself, $\alpha_{x,x} = 1 \neq 0$. This completes the proof. □

B.2 Proof of Observation 2.2

Assume that $\mathbf{z} = \mathbf{x}$ and all $\mathbf{x}$-variables are essential for h. Suppose, $A = \begin{bmatrix} A_{\mathbf{x}} & A_1 \\ A_2 & A_{\mathbf{y}} \end{bmatrix}$, where the rows and columns of $A_{\mathbf{x}}, A_{\mathbf{y}}$ are labelled by $\mathbf{x}$ and $\mathbf{y}$, respectively. It is sufficient to show that $A_1 = 0$. Let $g = h(A(\mathbf{x}, \mathbf{y})^T) \in \mathbb{F}[\mathbf{x}]$. Pretend that g, h are polynomials in $\mathbf{x} \uplus \mathbf{y}$. Let $\nabla g = ([\nabla g]_{\mathbf{x}}, [\nabla g]_{\mathbf{y}})^T$, where $[\nabla g]_{\mathbf{x}} = \left(\frac{\partial g}{\partial x}\right)_{x \in \mathbf{x}}$ and $[\nabla g]_{\mathbf{y}} = \left(\frac{\partial g}{\partial y}\right)_{y \in \mathbf{y}} = 0$. Similarly, let $\nabla h = ([\nabla h]_{\mathbf{x}}, [\nabla h]_{\mathbf{y}})^T$, where $[\nabla h]_{\mathbf{y}} = 0$. By the chain rule,

$$\nabla g = A^T \cdot [\nabla h](A(\mathbf{x}, \mathbf{y})^T). \tag{1}$$

Since $\mathbf{x}$ is the set of essential variables of h, the entries in $[\nabla h]_{\mathbf{x}}$ are $\mathbb{F}$-linearly independent, and thus, the entries in $[\nabla h]_{\mathbf{x}}(A(\mathbf{x}, \mathbf{y})^T)$ are also $\mathbb{F}$-linearly independent. Now, it is easy to see from Equation (1) and the structure of A that $A_1^T = 0$. Otherwise, we get a non-zero linear combination of $\frac{\partial h}{\partial x}(A(\mathbf{x}, \mathbf{y})^T), x \in \mathbf{x}$, which is equal to 0 (as $[\nabla g]_{\mathbf{y}} = [\nabla h]_{\mathbf{y}} = 0$), and this leads to a contradiction.

Now, suppose $\mathbf{x} \neq \mathbf{z}$. Let $P \in \mathrm{GL}(|\mathbf{x}| + |\mathbf{y}|, \mathbb{F})$ be a permutation matrix that maps $\mathbf{x}$ to $\mathbf{z}$, $\mathbf{z}$ to $\mathbf{x}$ and every other variable to itself. We know $h(A(\mathbf{x}, \mathbf{y})^T) \in \mathbb{F}[\mathbf{z}]$. Then, note that $h(AP(\mathbf{x}, \mathbf{y})^T) \in \mathbb{F}[\mathbf{x}]$. It follows from the above argument that AP maps every $\mathbf{x}$-variable to a linear form in $\mathbf{x}$. This implies A maps every $\mathbf{x}$-variable to a linear form in $\mathbf{z}$.

Now, suppose that not all $\mathbf{x}$-variables are necessarily essential for h. Let $C \in \mathrm{GL}(|\mathbf{x}| + |\mathbf{y}|, \mathbb{F})$ be such that $h(C(\mathbf{x}, \mathbf{y})^T)$ has no redundant variables. Then, Fact 2.2 implies that every truly essential variable of h is in $\mathrm{var}(h(C(\mathbf{x}, \mathbf{y})^T))$. The argument in the above paragraph implies that $C^{-1}A$ maps all variables in $\mathrm{var}(h(C(\mathbf{x}, \mathbf{y})^T))$ to linear forms in $\mathbf{z}$. Because C maps every truly essential variable of h to itself, A maps every truly essential variable to a linear form in $\mathbf{z}$. □

B.3 Proof of Observation 2.3

Suppose, $\sum_{x \in \mathbf{x}} \alpha_x \frac{\partial h}{\partial x} + \sum_{y \in \mathbf{y}} \beta_y \frac{\partial h}{\partial y} = 0$. Since $h = g(\mathbf{x})^e \cdot p(\mathbf{x}, \mathbf{y})$ and $e \geq 1$, we get

$$\sum_{x \in \mathbf{x}} \alpha_x \left(g^e \frac{\partial p}{\partial x} + e \cdot g^{e-1} \cdot p \frac{\partial g}{\partial x} \right) + \sum_{y \in \mathbf{y}} \beta_y g^e \frac{\partial p}{\partial y} = 0.$$

On dividing the above equation by g^{e-1} and rearranging the terms we get

$$g \left(\sum_{x \in \mathbf{x}} \alpha_x \frac{\partial p}{\partial x} + \sum_{y \in \mathbf{y}} \beta_y \frac{\partial p}{\partial y} \right) + e \cdot p \left(\sum_{x \in \mathbf{x}} \alpha_x \frac{\partial g}{\partial x} \right) = 0.$$

As g and p are coprime, and $\deg\left(\sum_{x \in \mathbf{x}} \alpha_x \frac{\partial g}{\partial x}\right) < \deg(g)$, we get $\sum_{x \in \mathbf{x}} \alpha_x \frac{\partial g}{\partial x} = 0$. But, this implies $\alpha_x = 0$ for every $x \in \mathbf{x}$, since $N_{ess}(g) = |\mathbf{x}|$. Hence, every $\mathbf{x}$-variable is truly essential for h. □

B.4 Proof of Observation 2.4

Let $\alpha_1 \frac{\partial h}{\partial x_1} + \alpha_2 \frac{\partial h}{\partial x_2} + \sum_{y \in \mathbf{y}} \beta_y \frac{\partial h}{\partial y} = 0$, where $\alpha_1, \alpha_2, \beta_y \in \mathbb{F}$ for $y \in \mathbf{y}$. Since $h = \sum_{i \geq 0} p_i(\mathbf{y})(x_1 x_2)^i$,

$$\alpha_1 \left(\sum_{i \geq 1} i \cdot p_i \cdot x_1^{i-1} x_2^i \right) + \alpha_2 \left(\sum_{i \geq 1} i \cdot p_i \cdot x_1^i x_2^{i-1} \right) + \sum_{y \in \mathbf{y}} \beta_y \left(\sum_{i \geq 0} (x_1 x_2)^i \frac{\partial p_i}{\partial y} \right) = 0.$$

Notice that $\alpha_1 \left(\sum_{i\geq 1} i \cdot p_i \cdot x_1^{i-1} x_2^i\right)$, $\alpha_2 \left(\sum_{i\geq 1} i \cdot p_i \cdot x_1^i x_2^{i-1}\right)$, and $\sum_{y\in \mathbf{y}} \beta_y \left(\sum_{i\geq 0} (x_1 x_2)^i \frac{\partial p_i}{\partial y}\right)$ are monomial disjoint. Thus, each of these three polynomials is zero. Suppose $\alpha_1 \neq 0$. Then, $\sum_{i\geq 1} i \cdot p_i \cdot x_1^{i-1} x_2^i = 0$. As $\mathrm{char}(\mathbb{F}) = 0$ or $> d$, we get $p_i = 0$ for every $i \geq 1$, which is a contradiction. Thus, $\alpha_1 = 0$. Similarly, $\alpha_2 = 0$. Hence, x_1 and x_2 are truly essential for h. □

B.5 Proof of Observation 2.5

For $j \in I$, let φ_j be the substitution map defined as: $\varphi_j(x_j) = -\alpha_j{}^{-1} \left(\sum_{i\in[n]\setminus\{j\}} \alpha_i x_i + \beta\right)$, and $\varphi(x) = x$ for every $x \in \mathbf{x} \setminus \{x_j\}$. For $j_1, j_2 \in I$, let $g_{j_1} := \varphi_{j_1}(g)$ and $g_{j_2} := \varphi_{j_2}(g)$. Observe that $g_{j_1} - g_{j_2} \in \langle \ell \rangle$, which implies $g_{j_1} = \varphi_{j_1}(g_{j_2})$, as $\varphi_{j_1}(\ell) = 0$ and g_{j_1} is x_{j_1}-free. Hence, by chain rule, $\frac{\partial g_{j_1}}{\partial x_i} = \varphi_{j_1}\left(\frac{\partial g_{j_2}}{\partial x_i}\right) - \alpha_{j_1}^{-1}\alpha_i \cdot \varphi_{j_1}\left(\frac{\partial g_{j_2}}{\partial x_{j_1}}\right)$. As g_{j_2} is x_{j_2}-free,

$$\frac{\partial g_{j_1}}{\partial x_{j_2}} = -\alpha_{j_1}^{-1}\alpha_{j_2} \cdot \varphi_{j_1}\left(\frac{\partial g_{j_2}}{\partial x_{j_1}}\right) \implies \frac{\partial g_{j_1}}{\partial x_i} - \alpha_{j_2}^{-1}\alpha_i \cdot \frac{\partial g_{j_1}}{\partial x_{j_2}} = \varphi_{j_1}\left(\frac{\partial g_{j_2}}{\partial x_i}\right).$$

Notice that the space spanned by $\left\{\frac{\partial g_{j_1}}{\partial x_i} - \alpha_{j_2}^{-1}\alpha_i \cdot \frac{\partial g_{j_1}}{\partial x_{j_2}} \;:\; x_i \in \mathbf{x}\right\}$ is W_{j_1}, which (by the above equations) is $U := \left\langle \varphi_{j_1}\left(\frac{\partial g_{j_2}}{\partial x_i}\right) \;:\; x_i \in \mathbf{x}\right\rangle$. As φ_{j_1} is linear, $\dim U \leq \dim W_{j_2}$, implying $\dim W_{j_1} \leq \dim W_{j_2}$. Similarly, we can show that $\dim W_{j_2} \leq \dim W_{j_1}$. Therefore, $\dim W_{j_1} = \dim W_{j_2}$.

B.6 Proof of Observation 2.6

From the definition of a formula, the first three properties of Definition 2.6 are satisfied by $\mathtt{C}$. We now "push" the labels on the edges of $\mathtt{C}$ down to the leaves so that the variables labelling the leaves are scaled. Then, we apply an invertible diagonal transformation S to $\mathbf{x}$ to rescale the variables appropriately. This ensures that property 4 is satisfied. To satisfy property 5, observe that if a $+$ gate has variable children $x_{i_1}, \ldots, x_{i_m}$ and constant children $\gamma_1, \ldots, \gamma_k$, then we can replace all the constants by $\gamma = \gamma_1 + \ldots + \gamma_k$, and apply an invertible affine transformation that maps x_{i_1} to $x_{i_1} - (x_{i_2} + \cdots + x_{i_m} + \gamma)$ and every other variable to itself.

Suppose u is a $+$ gate that has among its children a variable x and a $\times$ gate v such that v has two children – a variable y and a $+$ gate v'. Suppose v' has a constant child γ. The polynomial computed at v is of the form $x + y(T + \gamma) +$ other terms $= (x + \gamma y) + yT +$ other terms, where T is x and y free. Now, if we apply an invertible linear transformation that maps x to $x - \gamma y$ and every other variable to itself, then property 6 is satisfied with respect to nodes u and v. Finally, it is easy to see that this canonization process does not introduce any extra additive-constant.

B.7 Proof of Observation 2.7

Let $\mathrm{var}(\mathtt{C}) = \mathbf{x}$. Over any field, $N_{ess}(\mathtt{C}) \geq \dim \left\langle \frac{\partial \mathtt{C}}{\partial x} : x \in \mathbf{x} \right\rangle$. So it is sufficient to show that $\dim \left\langle \frac{\partial \mathtt{C}}{\partial x} : x \in \mathbf{x} \right\rangle = |\mathbf{x}|$. We will prove this by induction on the product depth Δ of $\mathtt{C}$. In the base case, $\Delta = 0$, and $\mathtt{C}$ computes a polynomial $x + \gamma$, for $\gamma \in \mathbb{F}$; so, $\dim \left\langle \frac{\partial \mathtt{C}}{\partial x} : x \in \mathbf{x} \right\rangle = |\mathbf{x}| = 1$. Suppose that the induction hypothesis holds for canonical ROFs of product depth $\Delta - 1$ or less.

Let $\mathtt{C} = T_1 + \ldots + T_s + \gamma$ be a canonical ROF of product depth Δ, where each T_i is a $\times$-rooted ROF having at least two non-constant, variable disjoint factors. Consider an $\mathbb{F}$-linear dependence $\sum_{x\in\mathbf{x}} \alpha_x \frac{\partial \mathtt{C}}{\partial x} = 0$, where $\alpha_x \in \mathbb{F}$. Then, $\sum_{x\in\mathrm{var}(T_i)} \alpha_x \frac{\partial T_i}{\partial x} \in \mathbb{F}$ for every $i \in [s]$. This is because T_i and T_j are variable disjoint for $i \neq j$. But $\sum_{x\in\mathrm{var}(T_i)} \alpha_x \frac{\partial T_i}{\partial x} \in \mathbb{F}$ implies $\sum_{x\in\mathrm{var}(T_i)} \alpha_x \frac{\partial T_i}{\partial x} = 0$, as T_i is a product of at least two non-constant factors and a common root of these variable disjoint factors is also a root of $\sum_{x\in\mathrm{var}(T_i)} \alpha_x \frac{\partial T_i}{\partial x}$. Now suppose $\alpha_x \neq 0$ for some $x \in \mathrm{var}(T_i)$ and $i \in [s]$. Let $T_i = Q_1 \cdots Q_m$, where $Q_1, \ldots, Q_m$ are variable disjoint $+$-rooted canonical ROFs of product depth at most $\Delta - 1$. Suppose that the x mentioned above is in $\mathrm{var}(Q_l)$. By the induction hypothesis, $\sum_{y\in\mathrm{var}(Q_l)} \alpha_y \frac{\partial Q_l}{\partial y} \neq 0$ unless every $\alpha_y = 0$. So the dependence $\sum_{x\in\mathrm{var}(T_i)} \alpha_x \frac{\partial T_i}{\partial x} = 0$ implies Q_l divides $\sum_{y\in\mathrm{var}(Q_l)} \alpha_y \frac{\partial Q_l}{\partial y} \neq 0$, which is not possible as the latter has a smaller degree. Therefore, $\alpha_x = 0$ for every $x \in \mathbf{x}$, and so, $\dim \left\langle \frac{\partial \mathtt{C}}{\partial x} : x \in \mathbf{x} \right\rangle = |\mathbf{x}|$.

B.8 Proof of Claim 2.1

Let $\mathbf{x} = \{x_1, \ldots, x_n\}$ be $\mathrm{var}(\mathtt{C})$, where $\mathtt{C} = T_1 + \cdots + T_s + \gamma$ is a canonical ROF. Let $\ell = \sum_{x\in\mathbf{x}} \alpha_x x + \alpha$, where either $|\mathrm{var}(\ell)| \geq 2$, $\alpha \in \mathbb{F}$ and for every $x \in \mathbf{x}$, $\alpha_x \in \mathbb{F}$ or $|\mathrm{var}(\ell)| = 1$ and $\alpha \in \mathbb{F}^\times$. Let $\mathbf{x}' = \mathbf{x} \setminus \{y_1\}$, $\ell_1 = \sum_{x\in\mathbf{x}'} -\alpha'_x x - \alpha'$, where for every $x \in \mathbf{x}', \alpha'_x = \alpha_x \alpha_{y_1}^{-1}$ and $\alpha' = \alpha\alpha_{y_1}^{-1}$. Notice that $\ell_1 \neq 0$. Then, we know that $\mathtt{C}_\ell = \mathtt{C}(y_1 = \ell_1, \mathbf{x}')$. As $\mathtt{C}$ is canonical, there exists at most one $l \in [s]$, such that T_l is a variable. If such an l exists and $\mathrm{var}(T_l) \cap \mathrm{var}(\ell) \neq \emptyset$ then we assume without loss of generality that $l = 1$. We also assume that $y_1 \in \mathrm{var}(T_1)$. Then, note that $\mathtt{C}_\ell = T'_1 + T_2 + \cdots + T_s + \gamma$, where $T'_1 := T_1(y_1 = \ell_1, \mathrm{var}(T_1) \setminus \{y_1\})$. For $l \in [2, s]$[36], let $\mathbf{x}'_l = \mathrm{var}(T_l)$ and $\mathbf{x}'_1 = \mathrm{var}(T_1) \setminus \{y_1\}$. We first prove the following two useful observations.

Observation B.1. $U := \left\langle \frac{\partial \mathtt{C}}{\partial x} : x \in \mathrm{var}(T_l), l \in [s], |\mathrm{var}(T_l)| \geq 2 \right\rangle$ *does not contain a non-zero constant.*

Proof. Suppose there exists an $\alpha \in U \cap \mathbb{F} \setminus \{0\}$. Let $\mathtt{C}' = \sum_{l\in[s],|\mathrm{var}(T_l)|\geq 2} T_l + \alpha y$, where y is a fresh variable. Then, $\mathtt{C}'$ is in the orbit of the canonical ROF $\sum_{l\in[s],|\mathrm{var}(T_l)|\geq 2} T_l + y$ and it follows from Fact 2.4 and Observation 2.7 that $N_{ess}(\mathtt{C}') = |\mathrm{var}(\mathtt{C}')|$. Thus, $W := \left\{ \frac{\partial \mathtt{C}'}{\partial x} : x \in \mathrm{var}(\mathtt{C}') \right\}$ is $\mathbb{F}$-linearly independent. Note that $U = \langle W \rangle$ but $\dim U < |W|$; a contradiction. So $U \cap \mathbb{F} \setminus \{0\} = \emptyset$. □

Observation B.2. *If* $|\mathrm{var}(T_1)| \leq 2$, *then* $N_{ess}(\mathtt{C}_\ell) \geq n - 2$.

Proof. There are two cases, $T_1 = y_1$ and $T_1 = y_1 y$ for some $y \in \mathbf{x}'$. For both cases, it follows from Observation 2.7 that $\left\{ \frac{\partial T_l}{\partial x} : l \in [2, s],\ x \in \mathrm{var}(T_l) \right\} \uplus \left\{ \frac{\partial T_1}{\partial y_1} \right\}$ is $\mathbb{F}$-linearly independent. Now, for any $l \in [2, s]$ and $x \in \mathrm{var}(T_l)$, $\frac{\partial \mathtt{C}_\ell}{\partial x} = \frac{\partial T_l}{\partial x} - \alpha'_x \frac{\partial T_1}{\partial y_1}$. Thus, $\left\{ \frac{\partial \mathtt{C}_\ell}{\partial x} : x \in \mathrm{var}(T_l), l \in [2, s] \right\}$ is $\mathbb{F}$-linearly independent. Hence, from Fact 2.1, when $T_1 = y_1$, $N_{ess}(\mathtt{C}_\ell) \geq n - 1$, and when $T_1 = y_1 y$, $N_{ess}(\mathtt{C}_\ell) \geq n - 2$. □

As $\mathtt{C}$ is multilinear, for every $x \in \mathbf{x}$, the individual degree of x in $\mathtt{C}$ is at most 2. Since $\mathrm{char}(\mathbb{F}) \neq 2$, for every $l \in [s], x \in \mathbf{x}'_l, \frac{\partial \mathtt{C}_\ell}{\partial x} \neq 0$. For $l \in [s], x \in \mathbf{x}'_l$, let $\beta_{l,x} \in \mathbb{F}$, such that

[36] For $m < n \in \mathbb{N}$, $[m, n] := \{m, \ldots, n\}$

$\sum_{l\in[s]}\sum_{x\in\mathbf{x}'_l}\beta_{l,x}\frac{\partial \mathtt{C}_\ell}{\partial x}=0$, which implies

$$\sum_{l\in[2,s]}\sum_{x\in\mathbf{x}'_l}\beta_{l,x}\left(\frac{\partial T_l}{\partial x}+\frac{\partial T'_1}{\partial x}\right)+\sum_{x\in\mathbf{x}'_1}\beta_{1,x}\frac{\partial T'_1}{\partial x}=0. \tag{2}$$

Let $I=\{l\in[s]:|\mathrm{var}(T_l)|=1\}$ and $J=[2,s]\setminus I$. As $\mathtt{C}$ is canonical, $|I|\le 1$. If $l\in I$, we call T_l as z. Now, we prove the claim by induction on the product-depth Δ of $\mathtt{C}$.

Base case: $\Delta=1$. Then, $\mathtt{C}_\ell=\ell_1T''_1+T_2+\cdots+T_s+\gamma$, where for every $l\in[2,s]$, T_l is a multilinear monomial and $T''_1=\prod_{x\in\mathbf{x}'_1}x$. Because of Observation B.2, we can assume that $z\notin\mathrm{var}(\ell_1)$. Thus Equation (2) becomes

$$\sum_{l\in J}\sum_{x\in\mathbf{x}'_l}\beta_{l,x}\left(-\alpha'_xT''_1+\frac{T_l}{x}\right)+\sum_{l\in I}\beta_{l,z}+\sum_{x\in\mathbf{x}'_1}\beta_{1,x}\left(-\alpha'_xT''_1+\ell_1\frac{T''_1}{x}\right)=0. \tag{3}$$

If $|\mathbf{x}'_1|\le 1$, we immediately have from Observation B.2 that $N_{ess}(C_\ell)\ge n-2$. So suppose that $|\mathbf{x}'_1|\ge 2$. Then for every $l\in J, x\in\mathbf{x}'_l$, the coefficient of $\frac{T_l}{x}$ in the above equation is $\beta_{l,x}$, which implies $\beta_{l,x}=0$. Also, as T''_1 and $\frac{T''_1}{x}$ are non-constant monomials for every $x\in\mathbf{x}'_1$ and as $|I|\le 1, \beta_{l,z}=0$. If $\mathrm{var}(\ell_1)\cap\mathrm{var}(T'_1)=\emptyset$, then Equation (3) becomes $\sum_{x\in\mathbf{x}'_1}\beta_{1,x}\cdot\ell_1\frac{T''_1}{x}=0$ which implies $\sum_{x\in\mathbf{x}'_1}\beta_{1,x}\cdot\frac{T''_1}{x}=0$. Then from Observation 2.7, $\beta_{1,x}=0$ for all $x\in\mathbf{x}'_1$ and $N_{ess}(\mathtt{C}_\ell)=n-1$. Otherwise pick any $y\in\mathrm{var}(\ell_1)\cap\mathrm{var}(T''_1)$ arbitrarily. Observe that the polynomial multiplied by y^2 in Equation (3) is $-\alpha'_y\sum_{x\in\mathbf{x}'_1\setminus\{y\}}\beta_{1,x}\frac{T''_1}{y\cdot x}$. As $\frac{T''_1}{y}$ is a canonical ROF, it follows from Observation 2.7 that $\beta_{1,x}=0$ for all $x\in\mathbf{x}\setminus y$. Then, from Equation (3) we have $\beta_{1,y}\left(-\alpha'_yT''_1+\ell_1\frac{T''_1}{y}\right)=0$. As the coefficient of T'' in this polynomial is $-2\alpha'_y\beta_{1,y}$, and $\mathrm{char}(\mathbb{F})\neq 2$, $\beta_{1,y}=0$. Hence, again $N_{ess}(\mathtt{C}_\ell)=n-1$. This proves the base case.

Induction step: Suppose $\Delta>1$ and the claim holds for all canonical ROFs of product-depth at most $\Delta-1$. Let $T_1=Q_1\cdots Q_m$, where for every $i\in[m]$, Q_i is either a variable or a +-rooted ROF. As in the base case, if $|\mathbf{x}'_1|\le 1$ or $z\in\mathrm{var}(\ell_1)$, then there is nothing to prove. So, suppose that $|\mathbf{x}'_1|\ge 2$ and $z\notin\mathrm{var}(\ell_1)$. We assume without loss of generality that $y_1\in\mathrm{var}(Q_1)$. It follows from the definition of $\mathtt{C}_\ell$ that $T'_1=Q'_1Q_2\cdots Q_m$, where $Q'_1=Q_1(y_1=\ell_1,\mathrm{var}(Q_1)\setminus\{y_1\})$. For $i\in[2,m]$, let $\widetilde{Q}_i=Q'_1\prod_{j\in[2,m]\setminus\{i\}}Q_j$ and $\widetilde{Q}_1=Q_2\cdots Q_m$. Let $\mathbf{x}'_{1,1}=\mathrm{var}(Q_1)\setminus\{y_1\}$ and for $i\in[2,m], \mathbf{x}'_{1,i}=\mathrm{var}(Q_i)$. For $i\in[m], x\in\mathbf{x}'_{1,i}$, rename the coefficient of $\frac{\partial T'_1}{\partial x}$ in Equation (2) as $c_{i,x}$. Then, Equation (2) becomes

$$\sum_{l\in J,x\in\mathbf{x}'_l}\beta_{l,x}\frac{\partial T_l}{\partial x}+\sum_{l\in I}\beta_{l,z}+\widetilde{Q}_1\left(\sum_{l\in J,x\in\mathbf{x}'_l}\beta_{l,x}\frac{\partial Q'_1}{\partial x}+\sum_{i\in[m],x\in\mathbf{x}'_{1,i}}c_{i,x}\frac{\partial Q'_1}{\partial x}\right)+\sum_{i\in[2,m]}\widetilde{Q}_i\left(\sum_{x\in\mathbf{x}'_{1,i}}c_{i,x}\frac{\partial Q_i}{\partial x}\right)=0. \tag{4}$$

Observation B.3. *If $T_1\neq y_1Q_2$, then for every $l\in J, x\in\mathbf{x}'_l, \beta_{l,x}=0$.*

Proof. If $m\ge 3$ then we substitute roots of Q_2 and Q_3 in Equation (4). As $|\mathbb{F}|>n$ and Q_2 and Q_3 are variable disjoint multilinear polynomials, roots of Q_2 and Q_3 exist over $\mathbb{F}$. Then, Observation 2.7 implies for $l\in J, x\in\mathbf{x}'_l, \beta_{l,x}=0$ and as $|I|\le 1, \beta_{l,z}=0$.

Now, suppose $m = 2$. Let the polynomial multiplied with $\widetilde{Q}_2 = Q'_1$ in Equation (4) be q_1. We plug in a root $\mathbf{a}$ of Q_2 in Equation (4). Let $h' = \sum_{l\in J, x\in \mathbf{x}'_l} \beta_{l,x} \frac{\partial T_l}{\partial x}$ and $h = h' + \sum_{l\in I} \beta_{l,z}$. Then $h = h(\mathbf{x}'_{1,2} = \mathbf{a}, \mathbf{x}' \setminus \mathbf{x}'_{1,2})$, and Equation (4) implies that $h = -q_1(\mathbf{a}) Q'_1(\mathbf{x}'_{1,2} = \mathbf{a}, \mathbf{x}' \setminus \mathbf{x}'_{1,2})$. Note that $q_1(\mathbf{a}) \in \mathbb{F}$. If either $q_1(\mathbf{a}) = 0$ or $\mathrm{var}(\ell_1) \cap \left(\uplus_{l\in J} \mathbf{x}'_l\right) = \emptyset$, then Observation B.1 implies that $h' = 0$. Otherwise, $\deg(Q'_1(\mathbf{x}'_{1,2} = \mathbf{a}, \mathbf{x}' \setminus \mathbf{x}'_{1,2})) = \deg(Q'_1)$. Also, in this case, $\deg(Q'_1) = \deg(Q_1)$. As $Q_1 \neq y_1$, $\deg(Q_1) \geq 2$. Hence, $\deg(Q'_1(\mathbf{x}'_{1,2} = \mathbf{a}, \mathbf{x}' \setminus \mathbf{x}'_{1,2})) \geq 2$. Then there exists a monomial p in $Q'_1(\mathbf{x}'_{1,2} = \mathbf{a}, \mathbf{x}' \setminus \mathbf{x}'_{1,2})$, such that $\deg(p) \geq 2$ and $\mathrm{var}(p) \cap \mathrm{var}(Q_1) \neq \emptyset$. Clearly, p is not in h, and as $h = -q_1(\mathbf{a}) Q'_1(\mathbf{x}'_{1,2} = \mathbf{a}, \mathbf{x}' \setminus \mathbf{x}'_{1,2})$, we get $h = 0$. This along with Observation B.1 implies $h' = 0$. Thus from Observation 2.7, $\beta_{l,x} = 0$ for every $l \in J, x \in \mathbf{x}'_l$. □

It follows from the above observation that when $T_1 \neq y_1 Q_2$, Equation (4) becomes

$$\sum_{l\in I} \beta_{l,z} + \widetilde{Q}_1 \left(\sum_{i\in[m], x\in \mathbf{x}'_{1,i}} c_{i,x} \frac{\partial Q'_1}{\partial x} \right) + \sum_{i\in[2,m]} \widetilde{Q}_i \left(\sum_{x\in \mathbf{x}'_{1,i}} c_{i,x} \frac{\partial Q_i}{\partial x} \right) = 0. \tag{5}$$

Now, we consider all the possible cases of T'_1. Recall $|\mathbf{x}'_1| \geq 2$, which implies that if $m = 2$ and Q'_1 is a linear polynomial then $\deg(Q_2) \geq 2$.

Case 1: $m = 2, \deg(Q_1) \geq 2$, and $Q_2 = y$ for some $y \in \mathbf{x}'$. Then, Equation (5) looks like

$$\sum_{l\in I} \beta_{l,z} + y \left(\sum_{x\in \mathbf{x}'_{1,1}} c_{1,x} \frac{\partial Q'_1}{\partial x} + c_{2,y} \frac{\partial Q'_1}{\partial y} \right) + Q'_1 c_{2,y} = 0. \tag{6}$$

If $y \notin \mathrm{var}(\ell_1)$, we put $y = 0$ in Equation (6). As $Q'_1(y = 0, \mathbf{x}' \setminus \{y\}) = Q'_1$, $c_{2,y} = 0$. As $|I| \leq 1$, $\beta_{l,z} = 0$. Thus we are left with $\sum_{x\in \mathbf{x}'_{1,1}} c_{1,x} \frac{\partial Q'_1}{\partial x} = 0$. Let $\mathbf{a} \in \mathbb{F}^{|\mathbf{x}' \setminus \mathbf{x}'_{1,1}|}$ be a point such that $\ell_1(\mathbf{x}' \setminus \mathbf{x}'_{1,1} = \mathbf{a}, \mathbf{x}'_{1,1}) \neq 0$; such a point exists. Notice that $N_{ess}(Q'_1(\mathbf{x}' \setminus \mathbf{x}'_{1,1} = \mathbf{a}, \mathbf{x}'_{1,1})) \leq N_{ess}(Q'_1)$. Because Q_1 is a product-depth $\Delta - 1$ ROF and $\ell_1(\mathbf{x}' \setminus \mathbf{x}'_{1,1} = \mathbf{a}, \mathbf{x}'_{1,1}) \neq 0$, it follows from the induction hypothesis that $N_{ess}(Q'_1(\mathbf{x}' \setminus \mathbf{x}'_{1,1} = \mathbf{a}, \mathbf{x}'_{1,1})) \geq |\mathrm{var}(Q_1)| - 2$. This means that at least $|\mathrm{var}(Q_1)| - 2$ many elements in $\left\{ \frac{\partial Q'_1}{\partial x} : x \in \mathbf{x}'_{1,1} \right\}$ are $\mathbb{F}$-linearly independent. Hence, at least $n - 2$ many elements in $\left\{ \frac{\partial \mathtt{C}_\ell}{\partial x} : x \in \mathbf{x}' \right\}$ are $\mathbb{F}$-linearly independent, and $N_{ess}(\mathtt{C}_\ell) \geq n - 2$.

If $y \in \mathrm{var}(\ell_1)$, $Q'_1 = yQ + q$ for some $Q \in \mathbb{F}[\mathbf{x}'_{1,1}]$ and $q \in \mathbb{F}[\mathbf{x}' \setminus \{y\}]$. If q is not a constant, just as before, we set $y = 0$ in Equation (6). This gives us $c_{2,y} = \beta_{l,z} = 0$. If $q \in \mathbb{F}$, note that $\mathbf{x}'_{1,1} \subseteq \mathrm{var}(Q)$, and hence y^2 divides $y \cdot \frac{\partial Q'_1}{\partial x}$ for all $x \in \mathbf{x}'_{1,1}$. This means that $\sum_{l\in I} \beta_{l,z} + c_{2,y} \left(y \frac{\partial Q'_1}{\partial y} + Q'_1 \right) = 0$. Now $y \frac{\partial Q'_1}{\partial y} = yQ$. Thus, $y \frac{\partial Q'_1}{\partial y} + Q'_1 = 2c_{2,y} yQ + q$. As $\mathrm{char}(\mathbb{F}) \geq 2$, $c_{2,y} = 0$, and thus $\beta_{l,z} = 0$. Then, using the induction hypothesis as before, we get $N_{ess}(\mathtt{C}_\ell) \geq n - 2$.

Case 2: $m = 2, Q_1 = y_1$, and $\deg(Q_2) \geq 2$. If $y_2 \in \mathrm{var}(\ell_1) \cap \mathbf{x}'_{1,2}$ then we redo this entire analysis by considering $\mathtt{C}_\ell = \mathtt{C}(y_2 = \ell_2, \mathbf{x} \setminus \{y_2\})$, where $\ell_2 := -\alpha_{y_2}^{-1}(\ell - \alpha_{y_2} y_2)$. Definition 2.3 and Observation 2.5 ensure that $N_{ess}(\mathtt{C}_\ell)$ is not affected by making this change to the definition of $\mathtt{C}_\ell$. Then, this case is same as Case 1 and we get the desired result. Otherwise, Equation (4) looks like

$$\sum_{l\in J, x\in \mathbf{x}'_l} \beta_{l,x} \frac{\partial T_l}{\partial x} + \sum_{l\in I} \beta_{l,z} + Q_2 \left(\sum_{l\in J, x\in \mathbf{x}'_l} -\beta_{l,x} \alpha'_x \right) + \ell_1 \left(\sum_{x\in \mathbf{x}'_{1,2}} c_{2,x} \frac{\partial Q_2}{\partial x} \right) = 0.$$

If Q_2 has a dangling variable connected to its top $+$ gate, let it be y_2. We shall consider the above equation without $c_{2,y}\frac{\partial Q_2}{\partial y_2}$. Observe that any monomial of the highest degree in Q_2 is not present in any other summand in the above equation. Hence $\sum_{l\in J, x\in \mathbf{x}'_l} -\beta_{l,x}\alpha'_x = 0$. Also, for every $x \in \mathbf{x}'_{1,2} \setminus \{y_2\}$, $\frac{\partial Q_2}{\partial x} \in \mathbb{F}[\mathbf{x}'_{1,2}]$. Hence, $\sum_{x\in \mathbf{x}'_{1,2}\setminus\{y_2\}} c_{2,x}\frac{\partial Q_2}{\partial x} = c$ for a $c \in \mathbb{F}$. It follows from Observation B.1 that $c = 0$, and hence from Observation 2.7 that $c_{2,x} = 0$ for all $x \in \mathbf{x}'_{1,2} \setminus \{y_2\}$. Observation B.1 and the fact that $|I| \leq 1$ imply that $\beta_{l,z} = 0$. Then, from Observation 2.7 $\beta_{l,x} = 0$ for all $l \in J$ and $x \in \mathbf{x}'_l$. Hence, $\left\{\frac{\partial \mathtt{C}_\ell}{\partial x} : x \in \mathbf{x}' \setminus \{y_2\}\right\}$ is $\mathbb{F}$-linearly independent and $N_{ess}(\mathtt{C}_\ell) \geq n-2$.

Case 3: $m = 2$, $\deg(Q_1) \geq 2$, and $\deg(Q_2) \geq 2$. In this case, Equation (5) becomes

$$\sum_{l\in I}\beta_{l,z} + Q_2\left(\sum_{i\in[2],x\in\mathbf{x}'_{1,i}} c_{i,x}\frac{\partial Q'_1}{\partial x}\right) + Q'_1\left(\sum_{x\in\mathbf{x}'_{1,2}} c_{2,x}\frac{\partial Q_2}{\partial x}\right) = 0.$$

Let the polynomials multiplied by Q'_1 and Q_2 in the above equation be q_1 and q_2, respectively. Let v be the parent of y_1 in $\mathtt{C}$ and path(v) be the path from the root of $\mathtt{C}$ to v. If v is the top-most + gate then substitute a root $\mathbf{a}$ of Q_2 in the above equation; $q_1(\mathbf{a}) \in \mathbb{F}$. As $\deg(Q'_1) \geq 2$ and $|I| \leq 1$, we get $\beta_{l,z} = 0$. Otherwise, there exists a $\times$ gate v' on path(v), such that $Q_{v',1}$ and $Q_{v',2}$ are children of v', where $Q_{v',1}$ lies on path(v) and $Q_{v',2}$ does not. Clearly, ℓ_1 is present in $Q_{v',1}$, and $Q_{v',2}$ is a +-rooted sub-ROF or a variable of Q_1. We first substitute a root $\mathbf{a}$ of $Q_{v',2}$ in the above equation and then plug in a root of $Q'_1(\mathbf{x}'_{1,2} = \mathbf{a}, \mathbf{x}' \setminus \mathbf{x}'_{1,2})$. In this process, note that $\mathbf{x}'_{1,2} \cup \mathrm{var}(\ell_1) \setminus \mathbf{x}'_{1,1}$ is untouched. As $|I| \leq 1$, $\beta_{l,z} = 0$. Further, since Q_2 is irreducible (Fact 2.5), we get that Q_2 either divides Q'_1 or q_1. As $\deg(Q_2) \geq 2$, Q_2 contains a monomial not present in Q'_1 and Q_2 does not divide Q'_1. As $\deg(Q_2) > \deg(q_1)$, Q_2 dividing q_1 implies that $q_1 = 0$. Thus, using Observation 2.7 we get that $c_{2,x} = 0$ for all $x \in \mathbf{x}'_{2,x}$. Then, using the induction hypothesis like in Case 1, we get $N_{ess}(\mathtt{C}_\ell) \geq n-2$.

Case 4: $m \geq 3$. By putting the roots of Q_2 and Q_3 in Equation (5), we get $\beta_{l,z} = 0$. For $i \in [2,m]$, let q_i be the polynomial multiplied with $\widetilde{Q}_i$ in Equation (5). Then for every $i \in [2,m]$, Q_i divides $\widetilde{Q}_i q_i$. As Q_i is irreducible (Fact 2.5), Q_i must divide q_i or $\widetilde{Q}_i$. Suppose there exists an i such that Q_i divides $\widetilde{Q}_i$. This happens if and only if $Q_i = x$ and $Q'_1 = \ell_1 = -\alpha'_x x$, where $x \in \mathbf{x}'$. Note that such an i is unique, say $i = 2$. Now, for every $j \in [3,m]$, Q_j must divide q_j. As $\deg(Q_j) > \deg(q_j)$, $q_j = 0$. Then, Equation (5) becomes $c_{2,x}(-\alpha_x\widetilde{Q}_1 + \widetilde{Q}_2) = -2c_{2,x}\alpha_x \cdot \prod_{j\in[3,m]} Q_j = 0$. As $\mathrm{char}(\mathbb{F}) \neq 2$ and $\alpha'_x \neq 0$, $c_{i,x} = 0$. If such an i does not exist, then $q_j = 0$ for all $j \in [2,m]$. In either case, using Observation 2.7 we get, $c_{j,x} = 0$ for every $j \in [2,m], x \in \mathbf{x}'_{1,j}$. Then, Equation (5) becomes $\sum_{x\in\mathbf{x}'_{1,1}} c_{1,x}\frac{\partial Q'_1}{\partial x} = 0$. Using the induction hypothesis like in Case 1, we get $N_{ess}(\mathtt{C}_\ell) \geq n-2$. □

B.9 Proof of Claim 2.3

Algorithm 8 Make-Polys-Var-Disjoint$(g_1, \ldots, g_m)$

Input: Black-box access to $g_1, \ldots g_m \in \mathbb{F}[\mathbf{x}]$ such that $N_{ess}(g_1 \cdots g_m) = N_{ess}(g_1) + \cdots + N_{ess}(g_m)$.
Output: An $A \in \mathrm{GL}(|\mathbf{x}|, \mathbb{F})$ such that $g_1(A\mathbf{x}), \ldots, g_m(A\mathbf{x})$ are pairwise variable disjoint and individually free of redundant variables.

1. $A \leftarrow I_{|\mathbf{x}|\times|\mathbf{x}|}$, $\mathbf{y} \leftarrow \emptyset$.

2. **for** $i = 1, \ldots, m$ **do**
3. $A_i \leftarrow$ Remove-Redundant-Vars$(g_i(A\mathbf{x}), \mathbf{y})$ (see Claim 2.2); $\mathbf{y}_i \leftarrow \text{var}(g_i(AA_i\mathbf{x}))$.
4. $A \leftarrow AA_i, \mathbf{y} \leftarrow \mathbf{y} \cup \mathbf{y}_i$.
5. **end for**
6. Return A.

The correctness of the algorithm follows from the observations below.

Observation B.4. *For every $i \in [m]$, $N_{ess}(g_1 \cdots g_i) = N_{ess}(g_1) + \cdots + N_{ess}(g_i)$.*

Proof. Follows from the fact that over any $\mathbb{F}$, $N_{ess}(h_1 h_2) \leq N_{ess}(h_1) + N_{ess}(h_2)$, for $h_1, h_2 \in \mathbb{F}[\mathbf{x}]$. □

Observation B.5. *Suppose $\mathbf{x} = \mathbf{y} \uplus \mathbf{z}$ and $h_1(\mathbf{y}), h_2(\mathbf{z}, \mathbf{y}) \in \mathbb{F}[\mathbf{x}]$ such that $N_{ess}(h_1) = |\mathbf{y}|$ and $N_{ess}(h_1 h_2) = N_{ess}(h_1) + N_{ess}(h_2)$. Then, $\mathbf{z}$ contains a set of essential variables of h_2.*

Proof. Observe that $\dim \left\langle \frac{\partial h_1 h_2}{\partial z} : z \in \mathbf{z} \right\rangle = \dim \left\langle \frac{\partial h_2}{\partial z} : z \in \mathbf{z} \right\rangle$; let this dimension be l. Then, by Fact 2.1, $N_{ess}(h_1 h_2) \leq l + |\mathbf{y}|$, which implies $N_{ess}(h_2) \leq l$ (as $N_{ess}(h_1) = |\mathbf{y}|$ and $N_{ess}(h_1 h_2) = N_{ess}(h_1) + N_{ess}(h_2)$). On the other hand, $\dim \left\langle \frac{\partial h_2}{\partial z} : z \in \mathbf{z} \right\rangle = l$ implies $N_{ess}(h_2) \geq l$. Hence, $N_{ess}(h_2) = l$, and so by Fact 2.1, $\mathbf{z}$ contains a set of essential variables of h_2. □

B.10 Proof of Claim 2.4

Algorithm 9 Make-Factors-Var-Disjoint($g(B\mathbf{x} + \mathbf{d})$)

Input: Black-box access to a $g(B\mathbf{x} + \mathbf{d}) \in \mathbb{F}[\mathbf{x}]_{\leq d}$, where $g = g_1 \cdots g_m$ for pairwise variable disjoint $g_1, \ldots, g_m$. ($B \in \text{GL}(|\mathbf{x}|, \mathbb{F})$, $\mathbf{d} \in \mathbb{F}^{|\mathbf{x}|}$, and $g, g_1, \ldots, g_m$ are unknown to the algorithm.)
Output: An $A \in \text{GL}(n, \mathbb{F})$ and a set V as stated in Claim 2.4.

1. Factorize $g(B\mathbf{x} + \mathbf{d})$ using Fact A.2. Let $F \leftarrow \{h_1, \ldots, h_e\}$ be the set of (black-boxes for the) irreducible factors of $g(B\mathbf{x} + \mathbf{d})$.
2. **while** $N_{ess}(\prod_{h \in F} h) \neq \sum_{h \in F} N_{ess}(h)$ **do** ,
3. For the *first* $l \in [|F|]$ s.t. $N_{ess}(h_1 \cdots h_l) \neq \sum_{j \in [l]} N_{ess}(h_j)$, find a $k \in [l-1]$ s.t. $N_{ess}(h_1 \cdots h_{k-1} \cdot h_l) = \sum_{j \in [k-1]} N_{ess}(h_j) + N_{ess}(h_l)$ but $N_{ess}(h_1 \cdots h_k \cdot h_l) \neq \sum_{j \in [k]} N_{ess}(h_j) + N_{ess}(h_l)$.
4. $F \leftarrow F \cup \{h_k \cdot h_l\}, F \leftarrow F \setminus \{h_k, h_l\}$. Rename the elements of F as $\{h_1, \ldots, h_s\}$.
5. **end while**
6. Let $F = \{h_1, \cdots, h_s\}$. $A \leftarrow$ Make-Polys-Var-Disjoint$(h_1, \ldots, h_s)$ (see Algorithm 8).
7. $V \leftarrow \{\text{var}(h_1(A\mathbf{x})), \ldots, \text{var}(h_s(A\mathbf{x}))\}$.
8. Return A and V.

The correctness of the algorithm follows from the following observation. Note that the number of essential variables of a polynomial can be computed efficiently using Claim 2.2. As we are merging factors in Step 4, it is clear that the running time of the algorithm is poly$(|\mathbf{x}|, d)$.

Observation B.6. *At Step 4, h_k and h_l are factors of $g_i(B\mathbf{x} + \mathbf{d})$ for some $i \in [m]$.*

Proof. For contradiction, suppose h_k is a factor of $g_i(B\mathbf{x} + \mathbf{d})$ and h_l is a factor of $g_j(B\mathbf{x} + \mathbf{d})$ for $i \neq j$. Let p be the product of all $h \in \{h_1, \ldots, h_{k-1}\}$ such that h is a factor of $g_i(B\mathbf{x} + \mathbf{d})$, q

the product of all $h \in \{h_1, \ldots, h_{k-1}\}$ such that h is a factor of $g_j(B\mathbf{x} + \mathbf{d})$, and r the product of all $h \in \{h_1, \ldots, h_{k-1}\}$ such that h is neither a factor of $g_i(B\mathbf{x} + \mathbf{d})$ nor a factor of $g_j(B\mathbf{x} + \mathbf{d})$. Then, $h_1 \cdots h_{k-1} = pqr$. Observe that $N_{ess}(pqrh_l) = N_{ess}(p) + N_{ess}(r) + N_{ess}(qh_l)$, as $g_1, \ldots, g_m$ are pairwise variable disjoint. On the other hand, from the condition $N_{ess}(h_1 \cdots h_{k-1} \cdot h_l) = N_{ess}(h_1) + \ldots N_{ess}(h_{k-1}) + N_{ess}(h_l)$ in Step 3, $N_{ess}(pqrh_l) = N_{ess}(p) + N_{ess}(q) + N_{ess}(r) + N_{ess}(h_l)$. Hence, $N_{ess}(qh_l) = N_{ess}(q) + N_{ess}(h_l)$. For a similar reason, $N_{ess}(ph_k) = N_{ess}(p) + N_{ess}(h_k)$. Now, $N_{ess}(pqrh_kh_l) = N_{ess}(ph_k) + N_{ess}(qh_l) + N_{ess}(r)$, as $g_1, \ldots, g_m$ are variable disjoint. This implies, $N_{ess}(pqrh_kh_l) = N_{ess}(p) + N_{ess}(h_k) + N_{ess}(q) + N_{ess}(h_l) + N_{ess}(r) = \sum_{j\in[k]} N_{ess}(h_j) + N_{ess}(h_l)$, which contradicts the condition $N_{ess}(h_1 \cdots h_k \cdot h_l) \neq N_{ess}(h_1) + \ldots N_{ess}(h_k) + N_{ess}(h_l)$ in Step 3. □

C Missing proofs from Section 3

C.1 Proof of Lemma 3.1

Structure of $\det(H_{\mathsf{C}})$**.** Notice that $\frac{\partial^2 \mathsf{C}}{\partial x \partial y} = 0$ if $x \in \mathrm{var}(T_k)$ and $y \in \mathrm{var}(T_{k'})$ for $k \neq k' \in [s]$. Thus H_{C} is a block diagonal matrix with the diagonal blocks being $H_{T_1}, \ldots, H_{T_s}$. Hence, $\det(H_{\mathsf{C}}) = \prod_{k\in[s]} \det(H_{T_k})$. So to prove that $\det(H_{\mathsf{C}}) \neq 0$, it suffices to show that $\det(H_{T_k}) \neq 0$ for all $k \in [s]$.

Lemma C.1. *Let $n \in \mathbb{N}$, $\mathbb{F}$ be a field with $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n$, and $\mathbf{x}$ be a variable set with $|\mathbf{x}| \leq n$. If T is a $\times$-rooted canonical ROF computing a polynomial in $\mathbb{F}[\mathbf{x}]$ of degree at least 2, then $\det(H_T) \neq 0$.*

Proof. We begin by developing an understanding of the entries of H_T. To do this, we first understand the derivatives of T. Let $\mathrm{path}(x)$ denote the path from the root of T to the leaf labelled by x. For an $x \in \mathbf{x}$, we define the *product-depth of x*, denoted by Δ_x, to be the number of $\times$ gates on $\mathrm{path}(x)$. We say that x is a *dangling variable* if x is directly connected to a $+$ gate. For an $x \in \mathbf{x}$, we *expand T along* $\mathrm{path}(x)$ as follows: let $T = Q_{x,1,1} \cdots Q_{x,1,m_1}$, and $x \in \mathrm{var}(Q_{x,1,1})$. Let $Q_{x,1,1} = T_{x,1,1} + \cdots + T_{x,1,s_1} + \gamma_1$, and $x \in \mathrm{var}(T_{x,1,1})$. Let l be any number less than $\Delta_x - 1$. After inductively defining $Q_{x,i,j}$ and $T_{x,i,j'}$ for all $i \in [l]$, $j \in [m_i]$, and $j' \in [s_i]$, let $T_{x,l,1} = Q_{x,l+1,1} \cdots Q_{x,l+1,m_{l+1}}$, with $x \in \mathrm{var}(Q_{x,l+1,1})$, and $Q_{x,l+1,1} = T_{x,l+1,1} + \cdots + T_{x,l+1,s_{l+1}} + \gamma_{l+1}$, with $x \in \mathrm{var}(T_{x,l+1,1})$. If x is not a dangling variable, let $T_{x,\Delta_x-1,1} = xQ_{x,\Delta_x,2} \cdots Q_{x,\Delta_x,m_{\Delta_x}}$ (here $Q_{x,\Delta_x,1} = x$). If x is a dangling variable, let $T_{x,\Delta_x-1,1} = Q_{x,\Delta_x,1} \cdots Q_{x,\Delta_x,m_{\Delta_x}}$ and $Q_{x,\Delta_x,1} = x + T_{x,\Delta_x,2} + \cdots + T_{x,\Delta_x,s_{\Delta_x}} + \gamma_{\Delta_x}$ (here $T_{x,\Delta_x,1} = x$). Then, $\frac{\partial T}{\partial x} = \prod_{i\in[\Delta_x]} \prod_{2\leq j\leq m_i} Q_{x,i,j}$.

The entries of H_T. For $x, y \in \mathbf{x}$, let $[H_T]_{x,y}$ denote the (x,y)-th enrty of H_T. Because T is multilinear, $[H_T]_{x,x} = 0$ for all $x \in \mathbf{x}$. For $x \neq y \in \mathbf{x}$, we define the *first common ancestor of x and y*, denoted by $\mathrm{fca}(x,y)$, to be the first gate that appears on both the path from the leaf labelled by x to the root of T as well as on the path from the leaf labelled by y to the root of T. There are two cases: $\mathrm{fca}(x,y)$ is a $+$ gate and $\mathrm{fca}(x,y)$ is a $\times$ gate. We now describe $\frac{\partial^2 T}{\partial x \partial y}$ in both these cases.

Observation C.1. *For all $x \neq y \in \mathbf{x}$ such that $\mathrm{fca}(x,y)$ is a $+$ gate, $[H_T]_{x,y} = [H_T]_{y,x} = 0$.*

Proof. Suppose that $\mathrm{fca}(x,y) = Q_{x,l,1}$ for some $1 \leq l \leq \min\{\Delta_x, \Delta_y\}$, and $y \in \mathrm{var}(T_{x,l,2})$. $\frac{\partial^2 T}{\partial x \partial y} = \frac{\partial^2 Q_{x,l,1}}{\partial x \partial y} \cdot \prod_{i\in[l]} \prod_{2\leq j\leq m_i} Q_{x,i,j} = \left(\frac{\partial^2 T_{x,l,1}}{\partial x \partial y} + \frac{\partial^2 T_{x,l,2}}{\partial x \partial y}\right) \cdot \prod_{i\in[l]} \prod_{2\leq j\leq m_i} Q_{x,i,j} = 0$. So, $[H_T]_{x,y} = [H_T]_{y,x} = 0$. □

The second case is when $\mathrm{fca}(x,y)$ is a $\times$ gate. Suppose that $\mathrm{fca}(x,y) = T$ or $\mathrm{fca}(x,y) = T_{x,l,1}$ for some $1 \leq l < \min\{\Delta_x, \Delta_y\}$. As we expanded T along $\mathrm{path}(x)$, we also expand it along

path(y) by defining $Q_{y,i,1}, \ldots, Q_{y,i,m'_i}$ for all $i \in [\Delta_y]$ and $T_{y,i,1}, \ldots, T_{y,i,s'_i}$ for all $i \in [\Delta_y]$ if y is a dangling variable, and for all $i \in [\Delta_y - 1]$ otherwise. Notice that for all $i \in [l]$, every $Q_{x,i,j} = Q_{y,i,j}$ and every $T_{x,i,j'} = T_{y,i,j'}$. Also, we can assume without loss of generality that $Q_{y,l+1,1} = Q_{x,l+1,2}$, $Q_{x,l+1,1} = Q_{y,l+1,2}$, and $Q_{x,l+1,j} = Q_{y,l+1,j}$ for all $3 \leq j \leq m_{l+1} = m'_{l+1}$. Let $\widetilde{Q}_{x,y} = \prod_{i \in [l]} \prod_{2 \leq j \leq m_i} Q_{x,i,j} \cdot \prod_{3 \leq j \leq m_{l+1}} Q_{x,l+1,j}$. Notice that $\widetilde{Q}_{x,y} = \prod_{i \in [l]} \prod_{2 \leq j \leq m'_i} Q_{y,i,j} \cdot \prod_{3 \leq j \leq m'_{l+1}} Q_{y,l+1,j}$. Then,

Observation C.2. *For all $x \neq y \in \mathbf{x}$ such that* fca(x, y) *is a $\times$ gate,*

$$[H_T]_{x,y} = [H_T]_{y,x} = \widetilde{Q}_{x,y} \prod_{l+1 < i \leq \Delta_x} \prod_{2 \leq j \leq m_i} Q_{x,i,j} \cdot \prod_{l+1 < i \leq \Delta_y} \prod_{2 \leq j \leq m'_i} Q_{y,i,j}.$$

Proof.

$$\begin{aligned}
\frac{\partial^2 T}{\partial x \partial y} &= \prod_{i \in [l]} \prod_{2 \leq j \leq m_i} Q_{x,i,j} \cdot \frac{\partial^2 Q_{x,l,1}}{\partial x \partial y} \\
&= \prod_{i \in [l]} \prod_{2 \leq j \leq m_i} Q_{x,i,j} \cdot \frac{\partial^2 T_{x,l,1}}{\partial x \partial y} \\
&= \prod_{i \in [l]} \prod_{2 \leq j \leq m_i} Q_{x,i,j} \cdot \prod_{3 \leq j \leq m_{l+1}} Q_{x,l+1,j} \cdot \frac{\partial Q_{x,l+1,1}}{\partial x} \cdot \frac{\partial Q_{y,l+1,1}}{\partial y} \\
&= \widetilde{Q}_{x,y} \prod_{l+1 < i \leq \Delta_x} \prod_{2 \leq j \leq m_i} Q_{x,i,j} \cdot \prod_{l+1 < i \leq \Delta_y} \prod_{2 \leq j \leq m'_i} Q_{y,i,j}.
\end{aligned}$$

□

Having gained an understanding of the entries of H_T, we proceed with the proof of the lemma. We shall call a $\times$-rooted canonical ROF a (Δ, m) ROF if it has product-depth Δ and has exactly m many product-depth $\Delta - 1$ ROFs connected to the top-most $\times$ gate. Let H'_T be the matrix obtained from H_T by taking x^{-1} common from the x-th row and the x-th column of H_T. Observe that for all $x, y \in \mathbf{x}$, $[H'_T]_{x,y} = xy \cdot [H_T]_{x,y}$. Also, notice that it suffices to show that $\det(H'_T) \neq 0$. We show this by induction on tuples of the form (Δ, m).

Base case. T is a $(1, m)$ ROF, where $2 \leq m \leq |\mathbf{x}|$. Then, T is a multilinear monomial, say $x_1 \cdots x_m$, and $\det(H'_T) = (-1)^{m-1}(m-1) \prod_{i \in [m]} x_i^m \neq 0$, as $\mathrm{char}(\mathbb{F}) = 0$ or $\geq n \geq |\mathbf{x}|$.

Induction step. T is a (Δ, m) ROF, for some $\Delta \geq 2$. Assume, by the way of induction, that $\det(H'_{T'}) \neq 0$ for all (Δ', m') ROFs T', where:

1. $\Delta' = 1$ and $m' \in \{2, \ldots, |\mathbf{x}|\}$, or
2. $1 < \Delta' < \Delta$ and $m' \in [|\mathbf{x}|]$, or
3. $\Delta = \Delta'$ and $m' < m$.

Pick a variable $x \in \mathrm{var}(T)$ as follows: arbitrarily pick a factor of T with product-depth exactly $\Delta - 1$. If there is no dangling variable inside this factor, then let x be any variable in it.

Otherwise let x be a dangling variable with the smallest product-depth in it. As before, we expand T along path(x) by defining $Q_{x,i,j}$ for all $i \in [\Delta_x]$ and $T_{x,i,j'}$ for all $i \in [\Delta_x]$ if x is a dangling variable, and for all $i \in [\Delta_x - 1]$ otherwise. Also, we assume without loss of generality that $Q_{x,1,1}, \ldots, Q_{x,1,m}$ are the only sub-ROFs of T with product-depth $\Delta - 1$. If x is not a dangling variable, let $\chi = \Delta_x - 1$; otherwise let $\chi = \Delta_x$. Let $U = \{y \in \text{var}(T) : \text{fca}(x, y) \text{ is a } \times \text{ gate}\} \uplus \{x\}$ and $\overline{U} = \text{var}(T) \setminus U = \{y \in \text{var}(T) : \text{fca}(x, y) \text{ is a } + \text{ gate}\}$. The following, easy to see observation gives a characterisation of U and $\overline{U}$.

Observation C.3. $U = \biguplus_{i \in [\Delta_x]} \biguplus_{2 \le j \le m_i} \text{var}(Q_{x,i,j}) \uplus \{x\}$ *and* $\overline{U} = \biguplus_{i \in [\chi]} \biguplus_{2 \le j \le s_i} \text{var}(T_{x,i,j})$.

We now upper bound the degree of x in $\det(H'_T)$, denoted by $\deg_x(\det(H'_T))$, in terms of $|U|$.

Observation C.4. $\deg_x(\det(H'_T)) \le |U|$.

Proof. $\det(H'_T) = \sum_{\sigma \in S_{\mathbf{x}}} (-1)^{sgn(\sigma)} \prod_{y \in \mathbf{x}} [H'_T]_{y,\sigma(y)}$, where $S_{\mathbf{x}}$ is the group of permutations of $\mathbf{x}$. It follows from Observations C.1 and C.2 that the only rows of H'_T containing x are the rows labelled by variables in U. Thus, for any $\sigma \in S_{\mathbf{x}}$, $[H'_T]_{y,\sigma(y)}$ contains x only if $y \in U$. Hence, at most $|U|$ many entries in $\left\{[H'_T]_{y,\sigma(y)} : y \in \mathbf{x}\right\}$ contain x. Also, the degree of x in each of those entries is at most 1. The observation follows. □

Let $N \subset S_{\mathbf{x}}$ be the set of all $\sigma \in S_{\mathbf{x}}$ such that the image of U under σ is U, and let $\overline{N} = S_{\mathbf{x}} \setminus N$.

Observation C.5. *For any* $\sigma \in \overline{N}$, $\deg_x \left(\prod_{y \in \mathbf{x}} [H'_T]_{y,\sigma(y)}\right) < |U|$.

Proof. As $\sigma \in \overline{N}$, there exists a $y' \in U$ such that $\sigma(y') \in \overline{U}$. It follows from Observations C.1 and C.2 that the only columns of H'_T containing x are the columns labelled by variables in U. Hence, $x \notin \text{var}\left([H'_T]_{y',\sigma(y')}\right)$. Then, even if all entries in $\left\{[H'_T]_{y,\sigma(y)} : y \ne y' \in U\right\}$ contain x, $\deg_x \left(\prod_{y \in \mathbf{x}} [H'_T]_{y,\sigma(y)}\right) < |U|$. □

Now,

$$\det(H'_T) = \sum_{\sigma \in N} (-1)^{sgn(\sigma)} \prod_{y \in \mathbf{x}} [H'_T]_{y,\sigma(y)} \; + \; \sum_{\sigma \in \overline{N}} (-1)^{sgn(\sigma)} \prod_{y \in \mathbf{x}} [H'_T]_{y,\sigma(y)}.$$

Let the first summand in the above expression be h. It follows from Observations C.4 and C.5 that to prove $\det(H'_T) \ne 0$, it suffices to show that $\deg_x(h) = |U|$.

Claim C.1. $h = \left(\sum_{\sigma_1 \in S_U} (-1)^{sgn(\sigma_1)} \prod_{y_1 \in U} [H'_T]_{y_1,\sigma_1(y_1)}\right) \cdot \left(\sum_{\sigma_2 \in S_{\overline{U}}} (-1)^{sgn(\sigma_2)} \prod_{y_2 \in \overline{U}} [H'_T]_{y_2,\sigma_2(y_2)}\right)$.

Proof. For any $\sigma \in S_{\mathbf{x}}$, let σ_1 be σ restricted to U and σ_2 be σ restricted to $\overline{U}$. For any $\sigma \in N$, notice that $\sigma_1 \in S_U$ and $\sigma_2 \in S_{\overline{U}}$. Thus,

$$\begin{aligned} h &= \sum_{\sigma \in N} (-1)^{sgn(\sigma)} \prod_{y \in \mathbf{x}} [H'_T]_{y,\sigma(y)} \\ &= \sum_{\substack{\sigma_1 \in S_U, \\ \sigma_2 \in S_{\overline{U}}}} (-1)^{sgn(\sigma_1)+sgn(\sigma_2)} \prod_{y_1 \in U} [H'_T]_{y_1,\sigma_1(y_1)} \cdot \prod_{y_2 \in \overline{U}} [H'_T]_{y_2,\sigma_2(y_2)} \end{aligned}$$

$$= \sum_{\sigma_1 \in S_U} (-1)^{sgn(\sigma_1)} \prod_{y_1 \in U} [H'_T]_{y_1,\sigma_1(y_1)} \left(\sum_{\sigma_2 \in S_{\overline{U}}} (-1)^{sgn(\sigma_2)} \prod_{y_2 \in \overline{U}} [H'_T]_{y_2,\sigma_2(y_2)} \right)$$

$$= \left(\sum_{\sigma_1 \in S_U} (-1)^{sgn(\sigma_1)} \prod_{y_1 \in U} [H'_T]_{y_1,\sigma_1(y_1)} \right) \cdot \left(\sum_{\sigma_2 \in S_{\overline{U}}} (-1)^{sgn(\sigma_2)} \prod_{y_2 \in \overline{U}} [H'_T]_{y_2,\sigma_2(y_2)} \right).$$

□

Observation C.6. $\deg_x \left(\sum_{\sigma_2 \in S_{\overline{U}}} (-1)^{sgn(\sigma_2)} \prod_{y_2 \in \overline{U}} [H'_T]_{y_2,\sigma_2(y_2)} \right) = 0.$

Proof. It follows from from Observations C.1 and C.2 that for no $y_2 \in \overline{U}$, does the row of H'_T labelled by y_2 contain x. □

Claim C.2. $\sum_{\sigma_2 \in S_{\overline{U}}} (-1)^{sgn(\sigma_2)} \prod_{y_2 \in \overline{U}} [H'_T]_{y_2,\sigma_2(y_2)} \neq 0.$

Proof. Notice that the given polynomial is the determinant of $[H'_T]_{\overline{U},\overline{U}}$, the sub-matrix of $[H'_T]$ whose rows and columns are labelled by variables in $\overline{U}$. We show that $[H'_T]_{\overline{U},\overline{U}}$ is a block diagonal matrix and all the diagonal blocks have non-zero determinant.

From Observation C.3, the rows and columns of $[H'_T]_{\overline{U},\overline{U}}$ are labelled by variables in $\biguplus_{i \in [\chi]} \biguplus_{2 \leq j \leq s_i}$ $\mathrm{var}(T_{x,i,j})$. We claim that $\frac{\partial^2 T}{\partial x_1 \partial x_2} = 0$ for all $x_1 \in \mathrm{var}(T_{x,i,j})$ and $x_2 \in \mathrm{var}(T_{x,i',j'})$, where $i \neq i'$ or $j \neq j'$. If $i = i'$, then both $T_{x,i,j}$ and $T_{x,i',j'}$ are children of the gate $Q_{x,i,1}$, and $\mathrm{fca}(x_1, x_2) = Q_{x,i,j}$. As $Q_{x,i,j}$ is a $+$ gate, from Observation C.1, $\frac{\partial^2 T}{\partial x_1 \partial x_2} = 0$. On the other hand, if $i \neq i'$, assume without loss of generality that $i < i'$. Then, observe that $T_{x,i',j'}$ is a sub-ROF of $T_{x,i,1}$. Thus $\mathrm{fca}(x_1, x_2)$ is again $Q_{x,i,j}$, and just as before $\frac{\partial^2 T}{\partial x_1 \partial x_2} = 0$. This implies that $[H'_T]_{\overline{U},\overline{U}}$ is a block diagonal matrix with diagonal blocks $[H'_T]_{\mathrm{var}(T_{x,i,j}),\mathrm{var}(T_{x,i,j})}$ for $i \in [\chi]$ and $2 \leq j \leq s_i$.

We now show that for all $i \in [\chi]$ and $2 \leq j \leq s_i$, the determinant of $[H'_T]_{\mathrm{var}(T_{x,i,j}),\mathrm{var}(T_{x,i,j})}$ is non-zero; this would prove the claim. Fix an $i \in [\chi]$ and a $j \in \{2, \ldots, s_i\}$. Observe that for any $x_1, x_2 \in \mathrm{var}(T_{x,i,j})$, $\frac{\partial^2 T}{\partial x_1 \partial x_2} = \frac{\partial^2 T_{x,i,j}}{\partial x_1 \partial x_2} \cdot \prod_{i' \in [i]} \prod_{2 \leq j' \leq m_{i'}} Q_{x,i',j'}$. So, $[H'_T]_{\mathrm{var}(T_{x,i,j}),\mathrm{var}(T_{x,i,j})} = \prod_{i' \in [i]} \prod_{2 \leq j' \leq m_{i'}} Q_{x,i',j'} \cdot [H'_{T_{x,i,j}}]$, and it is sufficient to prove that $\det([H'_{T_{x,i,j}}]) \neq 0$.[37] We claim that $T_{x,i,j}$ is not a single variable. The only way it can be a single variable is if it is a dangling variable. If x is not a dangling variable, then because of the way we picked x, there is no dangling variable inside $Q_{x,1,1}$. As $T_{x,i,j}$ is a sub-ROF of $Q_{x,1,1}$, it is not a dangling variable. Otherwise, as x is a dangling variable in $Q_{x,1,1}$ with the smallest product-depth, for all $i' \leq \Delta_x - 1$, and $2 \leq j' \leq s_{i'}$, $T_{x,i',j'}$ can not be a dangling variable. Also, x and $T_{x,\Delta_x,2}, \ldots, T_{x,\Delta_x,s_{\Delta_x}}$ are children of the same gate, viz. $Q_{x,\Delta_x,1}$. Because T is a canonical ROF, $T_{x,\Delta_x,2}, \ldots, T_{x,\Delta_x,s_{\Delta_x}}$ can not be dangling variables. Thus, $T_{x,i,j}$ is not a dangling variable, and is a (Δ', m') ROF for some $\Delta' < \Delta$ such that if $\Delta' = 1$, then $m' \geq 2$. Then it follows from the induction hypothesis that $\det([H'_{T_{x,i,j}}]) \neq 0$, proving the claim. □

[37] Since $T_{x,i,j}$ is a $\times$-rooted sub-ROF of T, we can define $[H'_{T_{x,i,j}}]$ in the same way as $[H'_T]$.

Because of Claim C.1, Observation C.6, and Claim C.2, to prove that $\deg_x(h) = |U|$, we only need to show that for $g := \sum_{\sigma_1 \in S_U} (-1)^{sgn(\sigma_1)} \prod_{y_1 \in U} [H'_T]_{y_1,\sigma_1(y_1)}$, $\deg_x(g) = |U|$. Let T' be the ROF obtained from T by replacing $T_{x,i,2} + \cdots + T_{x,i,s_i} + \gamma_i$ by 0 for all $i \in [\chi]$. Notice that $T' = x \prod_{i \in [\Delta_x]} \prod_{2 \le j \le m_i} Q_{x,i,j} = x \cdot \frac{\partial T}{\partial x}$. Hence, $\frac{\partial T}{\partial x} = \frac{\partial T'}{\partial x}$. Also, from Observation C.3, $\mathrm{var}(T') = U$.

Claim C.3. *When g and $\det(H'_{T'})$ are viewed as polynomials over $\mathbb{F}[\mathbf{x} \setminus \{x\}]$, the coefficient of $x^{|U|}$ is same in both the polynomials.*

Proof. $g = \sum_{\sigma_1 \in S_U} (-1)^{sgn(\sigma_1)} \prod_{y_1 \in U} [H'_T]_{y_1,\sigma_1(y_1)}$, $\det(H'_{T'}) = \sum_{\sigma_1 \in S_U} (-1)^{sgn(\sigma_1)} \prod_{y_1 \in U} [H'_{T'}]_{y_1,\sigma_1(y_1)}$ and for all $y_1, y_2 \in U$, $[H'_T]_{y_1,y_2}$ and $[H'_{T'}]_{y_1,y_2}$ are multilinear. Thus, it is sufficient to show that the coefficient of x is same in $[H'_T]_{y_1,y_2}$ and $[H'_{T'}]_{y_1,y_2}$ for all $y_1, y_2 \in U$. This is the same as showing that $\frac{\partial [H'_T]_{y_1,y_2}}{\partial x} = \frac{\partial [H'_{T'}]_{y_1,y_2}}{\partial x}$ for all $y_1, y_2 \in U$. There are three cases.

Case 1: Neither y_1 nor y_2 is x. Then,

$$\frac{\partial [H'_T]_{y_1,y_2}}{\partial x} = \frac{\partial}{\partial x}\left(y_1 y_2 \frac{\partial^2 T}{\partial y_1 \partial y_2}\right) = y_1 y_2 \frac{\partial^2}{\partial y_1 \partial y_2}\left(\frac{\partial T}{\partial x}\right) = y_1 y_2 \frac{\partial^2}{\partial y_1 \partial y_2}\left(\frac{\partial T'}{\partial x}\right) = \frac{\partial [H'_T]_{y_1,y_2}}{\partial x}.$$

Case 2: Exactly one of y_1 and y_2 is x; say $y_1 = x$. Then,

$$\frac{\partial [H'_T]_{y_1,y_2}}{\partial x} = \frac{\partial}{\partial x}\left(x y_2 \frac{\partial^2 T}{\partial x \partial y_2}\right) = y_2 \frac{\partial}{\partial y_2}\left(\frac{\partial T}{\partial x}\right) = y_2 \frac{\partial}{\partial y_2}\left(\frac{\partial T'}{\partial x}\right) = \frac{\partial}{\partial x}\left(x y_2 \frac{\partial^2 T'}{\partial x \partial y_2}\right) = \frac{\partial [H'_{T'}]_{y_1,y_2}}{\partial x}.$$

Case 3: $y_1 = y_2 = x$. In this case, both $[H'_T]_{y_1,y_2}$ and $[H'_{T'}]_{y_1,y_2}$ are 0. So, $\frac{\partial [H'_T]_{y_1,y_2}}{\partial x} = \frac{\partial [H'_{T'}]_{y_1,y_2}}{\partial x} = 0$. □

Now, every non-zero entry of $H'_{T'}$ contains x and the rows of $H'_{T'}$ are labelled by variables in U. Because we can take x common from all the rows of $H'_{T'}$, if $\det(H'_{T'}) \neq 0$, then $\deg_x(\det(H'_{T'})) = |U|$. Thus Claim C.3 implies that, $\deg_x(g) = |U|$ if and only if $\det(H'_{T'}) \neq 0$. Recall that T is a (Δ, m) ROF. If $m \geq 2$, then it follows from the definition of T' that it is a $(\Delta, m-1)$ ROF. Otherwise, if $m = 1$, i.e., if $Q_{x,i,1}$ is the only sub-ROF of T of product-depth Δ, then T' is a (Δ', m') ROF for some $\Delta' < \Delta$ and $m' \leq |\mathbf{x}|$. Also as T is a $\times$-rooted ROF, its fan-in, $m_1 \geq 2$. Thus, if $\Delta' = 1$, then $m' \geq 2$. So from the induction hypothesis, we have that $\det(H'_{T'}) \neq 0$. This proves the lemma. □

C.2 Proof of Claim 3.1

Let $T = Q \cdot Q_1 \cdots Q_m$ be a $\times$-rooted sub-ROF of $\mathtt{C}$. Let $\mathtt{C} = T_1 + \cdots + T_s + \gamma$. We saw in the proof of Lemma 3.1 that $\det(H_{\mathtt{C}}) = \prod_{k \in [s]} \det(H_{T_k})$. Thus, if T is a sub-ROF of T_k, then it is sufficient to show that Q^{e-1} is a factor of $\det(H_{T_k})$. Let $x \in \biguplus_{l \in [m]} \mathrm{var}(Q_l)$ and consider the x-th row of H_{T_k}. Like in the proof of Lemma 3.1, we expand T_k along the path(x). Then for any $y \in \mathrm{var}(T_k)$, $[H_{T_k}]_{x,y} = \frac{\partial}{\partial y}\left(\prod_{i \in [\Delta_x]} \prod_{2 \le j \le m_i} Q_{x,i,j}\right)$. Notice that for some $\lambda_x \in [\Delta_x]$ and $j \in [m_{\lambda_x}]$, say for $j = 2$, $Q = Q_{x,\lambda_x,2}$. Thus, Q is not a factor of $[H_{T_k}]_{x,y}$ only if $y \in \mathrm{var}(Q)$. For such a y,

$$[H_{T_k}]_{x,y} = \frac{\partial Q}{\partial y} \cdot \prod_{i \in [\lambda_x - 1]} \prod_{2 \le j \le m_i} Q_{x,i,j} \cdot \prod_{3 \le j \le m_{\lambda_x}} Q_{x,\lambda_x,j}. \tag{7}$$

We take Q common from every row of H_{T_k} labelled by variables in $\biguplus_{l\in[m]} \mathrm{var}(Q_l)$ to obtain a matrix H''_{T_k}. Now, $\det(H_{T_k}) = Q^e \det(H''_{T_k})$. So it suffices to show that $\det(H''_{T_k})$ is either a polynomial, or if it has a denominator, the denominator is just Q. Notice that the only entries of H''_{T_k} which are not polynomials but rational functions are $[H''_{T_k}]_{x,y}$, where $x \in \biguplus_{l\in[m]} \mathrm{var}(Q_l)$ and $y \in \mathrm{var}(Q)$. Let $\sigma \in S_{\mathrm{var}(T_k)}$ be any permutation that maps $x_1 \neq x_2 \in \biguplus_{l\in[m]} \mathrm{var}(Q_l)$ to $y_1 \neq y_2 \in \mathrm{var}(Q)$. Define $\sigma' \in S_{\mathrm{var}(T_k)}$ such that it maps x_1 to y_2, x_2 to y_1, and for all other $x \in \mathrm{var}(T_k)$, $\sigma'(x) = \sigma(x)$. Then,

$$\begin{aligned}
&(-1)^{sgn(\sigma)} \prod_{x\in\mathrm{var}(T_k)} [H''_{T_k}]_{x,\sigma(x)} - (-1)^{sgn(\sigma')} \prod_{x\in\mathrm{var}(T_k)} [H''_{T_k}]_{x,\sigma'(x)} \\
&= (-1)^{sgn(\sigma)} \prod_{x\in\mathrm{var}(T_k)\setminus\{x_1,x_2\}} [H''_{T_k}]_{x,\sigma(x)} \left([H''_{T_k}]_{x_1,y_1}[H''_{T_k}]_{x_2,y_2} - [H''_{T_k}]_{x_1,y_2}[H''_{T_k}]_{x_2,y_1} \right).
\end{aligned}$$

Now, expanding T along path(x_1) as well as path(x_2) we get,

$$\begin{aligned}
&Q^2 \left([H''_{T_k}]_{x_1,y_1}[H''_{T_k}]_{x_2,y_2} - [H''_{T_k}]_{x_1,y_2}[H''_{T_k}]_{x_2,y_1} \right) \\
&= \left(\frac{\partial Q}{\partial y_1} \cdot \prod_{i\in[\lambda_{x_1}-1]} \prod_{2\le j\le m_i} Q_{x,i,j} \cdot \prod_{3\le j\le m_{\lambda_{x_1}}} Q_{x_1,\lambda_{x_1},j} \right) \left(\frac{\partial Q}{\partial y_2} \cdot \prod_{i\in[\lambda_{x_2}-1]} \prod_{2\le j\le m'_i} Q_{x,i,j} \cdot \prod_{3\le j\le m'_{\lambda_{x_2}}} Q_{x_2,\lambda_{x_2},j} \right) \\
&- \left(\frac{\partial Q}{\partial y_2} \cdot \prod_{i\in[\lambda_{x_1}-1]} \prod_{2\le j\le m_i} Q_{x,i,j} \cdot \prod_{3\le j\le m_{\lambda_{x_1}}} Q_{x_1,\lambda_{x_1},j} \right) \left(\frac{\partial Q}{\partial y_1} \cdot \prod_{i\in[\lambda_{x_2}-1]} \prod_{2\le j\le m'_i} Q_{x,i,j} \cdot \prod_{3\le j\le m'_{\lambda_{x_2}}} Q_{x_2,\lambda_{x_2},j} \right) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(from Equation (7))} \\
&= 0.
\end{aligned}$$

Let U be the set of all permutations σ such that σ maps at most one variable in $\biguplus_{l\in[m]} \mathrm{var}(Q_l)$ to a variable in $\mathrm{var}(Q)$. Then,

$$\det[H''_{T_k}] = \sum_{\sigma\in U} (-1)^{sgn(\sigma)} \prod_{x\in\mathrm{var}(T_k)} [H''_{T_k}]_{x,\sigma(x)}.$$

As at most one of the $\left\{ [H''_{T_k}]_{x,\sigma(x)} : x \in \mathrm{var}(T_k) \right\}$ has a denominator and this denominator is Q, either $\det[H''_{T_k}]$ is a polynomial, or if it has a denominator, the denominator is just Q. Thus, Q^{e-1} is a factor of $\det(H_{T_k})$. □

C.3 Proof of Claim 3.2

From Lemma 3.1, $\det(H_{\mathtt{C}}) \neq 0$. Suppose that $x \in \mathrm{var}(\mathtt{C})$ is not a variable in the quadratic form of the top-most $+$ gate of $\mathtt{C}$, nor a dangling variable along some skewed path, nor a variable appearing in a quadratic form along some skewed path. There there exists a $\times$-gate T on the path from the root of $\mathtt{C}$ to the leaf labelled by x such that if $T = Q_1 \cdots Q_{m'}$, then $x \in \mathrm{var}(Q_1)$ and $|\mathrm{var}(Q_2)| + \cdots + |\mathrm{var}(Q_{m'})| \geq 2$. Then, Claim 3.1 implies that Q_1 is a factor of $\det(H_{\mathtt{C}})$. Now Q_1 is either a variable or a $+$-rooted sub-ROF, and therefore is irreducible (Fact 2.5). Then, the claim immediately follows from Observation 2.3. □

C.4 Proof of Claim 3.3

Suppose that $y_1 \in \mathbf{y}$ is present in $\det(H_{\mathtt{C}})$. Let the quadratic form of Q be $y_1y_2 + \ldots + y_{l-1}y_l$. Let P be a permutation matrix acting on $\mathbf{x} := \mathrm{var}(\mathtt{C})$ such that P maps y_1 to y_2, y_2 to y_1, and every other variable to itself. As, $\mathtt{C} = \mathtt{C}(P\mathbf{x})$, $\det\left(H_{\mathtt{C}}\right) = \det\left(H_{\mathtt{C}(P\mathbf{x})}\right)$. Also, from Fact 2.8, $\det\left(H_{\mathtt{C}(P\mathbf{x})}\right) = \det\left(H_{\mathtt{C}}\right)(P\mathbf{x})$. As y_1 is present in $\det\left(H_{\mathtt{C}}\right)$, y_2 is present in $\det\left(H_{\mathtt{C}}\right)(P\mathbf{x}) = \det\left(H_{\mathtt{C}}\right)$. For any odd $i \leq l-1$, let P be a permutation matrix mapping y_1 to y_i, y_2 to y_{i+1}, y_i to y_1, y_{i+1} to y_2, and all other variables to themselves. Again $\det\left(H_{\mathtt{C}}\right) = \det\left(H_{\mathtt{C}(P\mathbf{x})}\right)$ and $\det\left(H_{\mathtt{C}(P\mathbf{x})}\right) = \det\left(H_{\mathtt{C}}\right)(P\mathbf{x})$. As y_1 and y_2 appear in $\det\left(H_{\mathtt{C}}\right)$, y_i and y_{i+1} also appear in $\det\left(H_{\mathtt{C}}\right)(P\mathbf{x}) = \det\left(H_{\mathtt{C}}\right)$.

For any odd $i \leq l-1$, let S be a scaling matrix mapping y_i to $2y_i$, y_{i+1} to $\frac{y_{i+1}}{2}$ and every other variable to itself. $\mathtt{C} = \mathtt{C}(S\mathbf{x})$, and hence $\det\left(H_{\mathtt{C}}\right) = \det\left(H_{\mathtt{C}(S\mathbf{x})}\right)$. Also, from Fact 2.8, $\det\left(H_{\mathtt{C}(S\mathbf{x})}\right) = \det(H_{\mathtt{C}})(S\mathbf{x})$. Consider a monomial μ of $\det\left(H_{\mathtt{C}}\right)$ in which the degree of y_i is d_i, that of y_{i+1} is d_{i+1}, and whose coefficient is β. In $\det\left(H_{\mathtt{C}}\right)(S\mathbf{x})$, the coefficient of μ is $\beta \cdot 2^{d_i - d_{i+1}}$. Thus, $d_i = d_{i+1}$. Then from Observation 2.4 y_i and y_{i+1} are truly essential for $\det\left(H_{\mathtt{C}}\right)$. □

C.5 Proof of Claim 3.4

As argued in Section C.1, $\det(H_{\mathtt{C}}) = \prod_{k\in[s]} \det\left(H_{T_k}\right)$. It is sufficient to show that y is not present in $\det\left(H_{T_k}\right)$. It follows from Observations C.1 and C.2, that y does not appear in any entry of H_{T_k} because the only $x' \in \mathbf{x}$ for which $\frac{\partial^2 T_k}{\partial x' \partial y} \neq 0$ is x. But $\frac{\partial^2 T_k}{\partial x' \partial y} = 1$. Hence, $y \notin \mathrm{var}\left(\det\left(H_{T_k}\right)\right)$. □

D Missing proofs from Section 4

D.1 Proof of Claim 4.1

$[BR]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}} = [B]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}}\,[R]_{\mathbf{x},\mathbf{x}\setminus\{u_0\}}$, where $[B]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}}$ is the sub-matrix of B whose rows and columns are labelled by variables in $\mathbf{x}\setminus\{u_0\}$ and $\mathbf{x}$, respectively, while $[R]_{\mathbf{x},\mathbf{x}\setminus\{u_0\}}$ is the sub-matrix of R whose rows and columns are labelled by variables in $\mathbf{x}$ and $\mathbf{x}\setminus\{u_0\}$, respectively. For $x \in \mathbf{x}\setminus\{u_0\}$, let ℓ_x be the linear form that x is mapped to by B. Let $R = (r_{x,x'})_{x,x'\in\mathbf{x}}$. Then the (x,x')-the entry of $[BR]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}}$ is $\ell_x(\mathbf{r}_{x'})$, where $\mathbf{r}_{x'} = \{r_{x,x'} : x \in \mathbf{x}\}$. As B is invertible, $\{\ell_x(\mathbf{x}) : x \in \mathbf{x}\setminus\{u_0\}\}$ is linearly independent. Thus, the columns of $[BR]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}}$ are evaluations of linearly independent, degree 1 polynomials at independently chosen random points from F^n, where $|F| \geq n^5$. It is well known (see for instance Claim 2.2 of [KNS19]) that any such matrix is invertible with probability at least $1 - \frac{1}{n^4}$. □

D.2 Proof of Claim 4.2

Observe that H_2 is the Hessian of $\sum_{k\in[s']} T_k + \gamma$. Then $H_{\mathtt{C}} = \begin{bmatrix} H_2 & 0 \\ 0 & 0 \end{bmatrix}$. Fact 2.6 implies that $H_{f(R\mathbf{x})} = (BR)^T \cdot H_{\mathtt{C}}(BR\mathbf{x}+\mathbf{d}) \cdot (BR)$. It is easy to that $H_1 = [BR]^T_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}} \cdot H_2(BR\mathbf{x}+\mathbf{d}) \cdot [BR]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}}$, which implies $h = \det(H_1) = \beta^2 \det(H_2)(BR\mathbf{x}+\mathbf{d})$, where β is the determinant of $[BR]_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}}$. From Lemma 3.1, $\det(H_2) \neq 0$ and from Claim 4.1 $\beta \neq 0$ with high probability. Hence, h is also non-zero with high probability. Also, $u_0 \notin \mathrm{var}(\det(H_2))$ and hence $N_{ess}(\det(H_2)) \leq n-1$. Now,

$[\nabla h]_{\mathbf{x}\setminus\{u_0\}} = [BR]^T_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}} \cdot [\nabla \det(H_2)]_{\mathbf{x}\setminus\{u_0\}} (BR\mathbf{x}+\mathbf{d})$, where $[\nabla h]_{\mathbf{x}\setminus\{u_0\}}$ and $[\nabla \det(H_2)]_{\mathbf{x}\setminus\{u_0\}}$ are the gradient vectors of h and $\det(H_2)$ restricted to the entries corresponding to variables in $\mathbf{x}\setminus\{u_0\}$. From Claim 4.1, $[BR]^T_{\mathbf{x}\setminus\{u_0\},\mathbf{x}\setminus\{u_0\}}$ is invertible with high probability. So the spaces $\left\langle \frac{\partial h}{\partial x} : x \in \mathbf{x}\setminus\{u_0\} \right\rangle$ and $\left\langle \frac{\partial}{\partial x}\det(H_2) : x \in \mathbf{x}\setminus\{u_0\} \right\rangle$ have the same dimension with high probability. Then Facts 2.3 and 2.1 imply that a subset of $\mathbf{x}\setminus\{u_0\}$ contains a set of essential variables of h with high probability. Thus, u_0 is redundant for h with high probability. □

D.3 Proof of Claim 4.3

$\widehat{T}_k(R\mathbf{x}) = T_k(BR\mathbf{x}+\mathbf{d})$ implies that $g_k(\mathbf{x}) = g'_k(C_k^{-1}BRA_0\mathbf{x} + C_k^{-1}\mathbf{d})$. As $\mathrm{var}(g_k) = \mathrm{var}(g'_k) = \mathbf{z}_k$ and none of them have any redundant variables, $\mathbf{z}_k$ are the truly essential variables of g and g'. Thus Observation 2.2 implies that $C_k^{-1}BRA_0$ maps every $z \in \mathbf{z}_k$ to a linear form in $\mathbf{z}_k$. Also, from Fact 2.2 we have that C_k maps every $z \in \mathbf{z}'_k$ and every $y \in \mathbf{y}_k$ to itself, and every $z \in \mathbf{z}''_k$ to a linear form that looks like $z + \sum_{y\in\mathbf{y}_k\cap\mathrm{var}(h'_k)} \alpha_y y$. Multiplying C_k to $C_k^{-1}BRA_0$ yields the claim. □

D.4 Proof of Claim 4.4

BA_0 is invertible and from Claim 4.3, for every $z \in \mathbf{z}'$, $\left[\ell_z^{(0)}\right]_{\mathbf{y}} = 0$. Hence, the sub-matrix $[BA_0]_{\mathbf{z}''\uplus\mathbf{y},\mathbf{y}}$ of BA_0 containing rows corresponding to variables in $\mathbf{z}''\uplus\mathbf{y}$ and columns corresponding to variables in $\mathbf{y}$ is full rank. From Claim 4.3, we have that all rows of $[BA_0]_{\mathbf{z}'',\mathbf{y}}$ are in the $\mathbb{F}$-span of the rows of $[BA_0]_{\mathbf{y},\mathbf{y}}$. Thus $[BA_0]_{\mathbf{y},\mathbf{y}}$ is full rank. The claim follows by noticing that the entries of $[BA_0]_{\mathbf{y},\mathbf{y}}\,\mathbf{y}$ are exactly the linear forms $\left[\ell_y^{(0)}\right]_{\mathbf{y}}$ for all $y \in \mathbf{y}$. □

D.5 Proof of Claim 4.5

Observe that for every $i \in [m]$, the i-th iteration of the loop only works with $\widetilde{q}_i(A'_1\mathbf{x})$ (where A'_1 is as in Step 4) and computes a C_i which only acts non-trivially on $\mathrm{var}\left(\widetilde{q}_i(A'_1\mathbf{x})\right)$. Thus, we can analyse every iteration of the loop in isolation, and it sufficient to prove that after the i-th iteration,

$$\widehat{q}_i(A'_1C_i\mathbf{x}) = \left(y'_{i,1,1}+h_{i,1,1}\right)\left(y'_{i,1,2}+h_{i,1,2}\right) + \cdots + \left(y'_{i,m_i,1}+h_{i,m_i,1}\right)\left(y'_{i,m_i,2}+h_{i,m_i,2}\right).$$

Fix an $i \in \{0,\ldots,m\}$ and let $\mathrm{var}\left(\widetilde{q}_i(A'_1\mathbf{x})\right) = \left\{y'_{i,1,1},\, y'_{i,1,2},\ldots,y'_{i,m_i,1},\, y'_{i,m_i,2}\right\}$. As mentioned before, $\widetilde{q}_i(A'_1\mathbf{x})$ has no redundant variables. Thus, $C_i \in \mathrm{GL}(2m_i,\mathbb{F})$ output by the QFE algorithm is such that after it has been extended to map every variable in $\mathbf{x}\setminus\mathrm{var}\left(\widetilde{q}_i(A'_1\mathbf{x})\right)$ to itself, $\widetilde{q}_i(A'_1C_i\mathbf{x}) = y'_{i,1,1}y'_{i,1,2} + \cdots + y'_{i,m_i,1}y'_{i,m_i,2}$.

For all $j \in [m_i]$ and $l \in [2]$, let $\alpha_{l,j}$ be the $y_{i,j,l}$-th entry of $B\mathbf{b}+\mathbf{d}$ and $p_{j,l} = \left[\ell^{(0)}_{y_{i,j,l}}\right]_{\mathbf{z}} + \alpha_{l,j}$. Then,

$$\sum_{j\in[m_i]}\left(\left[\ell^{(0)}_{y_{i,j,1}}\right]_{\mathbf{y}} + p_{j,1}\right)\left(\left[\ell^{(0)}_{y_{i,j,2}}\right]_{\mathbf{y}} + p_{j,2}\right)(A'_1C_i\mathbf{x}) = \sum_{j\in[m_i]}\left(\ell_{j,1}+p_{j,1}\right)\left(\ell_{j,2}+p_{j,2}\right),$$

where for $j \in [m_i]$, $l \in [2]$, $\ell_{j,l} := \left[\ell^{(0)}_{y_{i,j,1}}\right]_{\mathbf{y}}(A'_1C_i\mathbf{x})$. Since $A'_1C_i \in \mathrm{GL}(n,\mathbb{F})$, Claim 4.4 implies that $\left\{\ell_{j,l} : j \in [m_i],\, l \in [2]\right\}$ is linearly independent. Now $\widetilde{q}_i(A'_1C_i) = \sum_{j\in[m_i]}\ell_{j,1}\ell_{j,2}$ and $\widetilde{q} \in \mathrm{orb}(q_i)$.

Also neither q_i nor $\widetilde{q}_i$ have any redundant variables. Hence from Observation 2.2, for all $j \in [m_i]$ and $l \in [2]$, $\ell_{j,l}$ is a linear form solely in $\left\{y'_{i,1,1}, y'_{i,1,2}, \ldots, y'_{i,m_i,1}, y'_{i,m_i,2}\right\}$. Expanding the right hand side of the above equation,

$$\sum_{j\in[m_i]} (\ell_{j,1} + p_{j,1})(\ell_{j,2} + p_{j,2}) = \sum_{j\in[m_i]} \ell_{j,1}\ell_{j,2} + \sum_{j\in[m_i]} (\ell_{j,1}p_{j,2} + \ell_{j,2}p_{j,1}) + \sum_{j\in[m_i]} p_{j,1}p_{j,2}. \tag{8}$$

For $j \in [m_i]$, let $h_{i,j,1}$ and $h_{i,j,2}$ be the coefficients of $y_{i,j,2}$ and $y_{i,j,1}$ in $\sum_{j\in[m_i]}(\ell_{j,1}p_{j,2} + \ell_{j,2}p_{j,1})$ respectively. Then, $h_{i,j,1}, h_{i,j,2} \in \mathbb{F}[\mathbf{z}]$ are linear polynomials and $\sum_{j\in[m_i]}(\ell_{j,1}p_{j,2} + \ell_{j,2}p_{j,1}) = \sum_{j\in[m_i]}(y_{i,j,1}h_{i,j,2} + y_{i,j,2}h_{i,j,1})$. Now, $\sum_{j\in[m_i]} \ell_{j,1}\ell_{j,2} = \sum_{j\in[m_i]} y_{i,j,1}y_{i,j,2}$. Putting these in equation (8),

$$\sum_{j\in[m_i]} (\ell_{j,1} + p_{j,1})(\ell_{j,2} + p_{j,2}) = \sum_{j\in[m_i]} (y_{i,j,1} + h_{i,j,1})(y_{i,j,2} + h_{i,j,2}) + \sum_{j\in[m_i]} (p_{j,1}p_{j,2} - h_{i,j,1}h_{i,j,2}) \tag{9}$$

Substitute $y_{i,j,l} = y_{i,j,l} - h_{i,j,l}$ for every $j \in [m_i], l \in [2]$ in the above equation. Then we get

$$\sum_{j\in[m_i]} (\ell_{j,1} + p'_{j,1})(\ell_{j,2} + p'_{j,2}) = \sum_{j\in[m_i]} y_{i,j,1}y_{i,j,2} + \sum_{j\in[m_i]} (p_{j,1}p_{j,2} - h_{i,j,1}h_{i,j,2}),$$

where for every $j \in [m_i]$, $l \in [2]$, $p'_{j,l} \in \mathbb{F}[\mathbf{z}]$ is a linear polynomial. Note that the right hand side of the above equation does not have a monomial containing variables from both $\mathbf{y}$ and $\mathbf{z}$. Thus we get $\sum_{j\in[m_i]}(\ell_{j,1}p'_{j,2} + \ell_{j,2}p'_{j,1}) = 0$. Since $\{\ell_{j,l} : j \in [m_i],\ l \in [2]\}$ is linearly independent, it is easy to see that for every $j \in [m_i]$, $p'_{j,1} = p'_{j,2} = 0$, which implies $\sum_{j\in[m_i]}(p_{j,1}p_{j,2} - h_{i,j,1}h_{i,j,2}) = 0$. Hence,

$$\begin{aligned}
\widehat{q}_i(A'_1C_i\mathbf{x}) &= \sum_{j\in[m_i]} \left(\left[\ell^{(0)}_{y_{i,j,1}}\right]_{\mathbf{y}} + p_{j,1}\right)\left(\left[\ell^{(0)}_{y_{i,j,2}}\right]_{\mathbf{y}} + p_{j,2}\right)(A'_1C_i\mathbf{x}) \\
&= \sum_{j\in[m_i]} (\ell_{j,1} + p_{j,1})(\ell_{j,2} + p_{j,2}) \\
&= \sum_{j\in[m_i]} (y_{i,j,1} + h_{i,j,1})(y_{i,j,2} + h_{i,j,2}) \qquad \text{(from Equation (9))}.
\end{aligned}$$

□

D.6 Proof of Claim 4.6

$\left\{\ell^{(1)}_x : x \in \mathbf{x}\right\}$ is linearly independent. As A'_1 acts as identity on $\mathbf{z} \uplus \mathbf{u}$ (Observation 4.1), from Claim 4.3, we get that $\ell^{(1)}_z \in \mathbb{F}[\mathbf{z}]$ for all $z \in \mathbf{z}'$. Hence, $\dim\left\langle \left[\ell^{(1)}_x\right]_{\mathbf{y}} : x \in \mathbf{z}'' \uplus \mathbf{y} \right\rangle = |\mathbf{y}|$. From Claim 3.3, no $y \in \mathbf{y} \setminus \mathbf{u}$ is in $\mathrm{var}(\det(H_2))$. Thus, by applying A'_1 on both sides of the equation in the second point of Claim 4.3, we get that for all $z \in \mathbf{z}''$, $\ell^{(1)}_z = \ell'_z + \sum_{u\in\mathbf{u}} \alpha_u \ell^{(1)}_u$, where $\ell'_z \in \mathbb{F}[\mathbf{z}]$. Hence, $\left\{\left[\ell^{(1)}_z\right]_{\mathbf{y}} : z \in \mathbf{z}''\right\} \in \mathbb{F}\text{-span}\left\{\left[\ell^{(1)}_y\right]_{\mathbf{y}} : y \in \mathbf{y}\right\}$; so $\left\{\left[\ell^{(1)}_y\right]_{\mathbf{y}} : y \in \mathbf{y}\right\}$ is linearly independent. Now, $\left\{\left[\ell^{(1)}_y\right]_{\mathbf{y}} : y \in \mathbf{y} \setminus \mathbf{u}\right\} = \mathbf{y} \setminus \mathbf{u}$. Thus, $\left\{\left[\ell^{(1)}_u\right]_{\mathbf{u}} : u \in \mathbf{u}\right\}$ is linearly independent.

D.7 Proof of Claim 4.7

Immediately after Step 8 of Procedure 2 is executed, $\det(H_2)(BRA_0\mathbf{x} + \mathbf{d}) = h(A_0\mathbf{x}) \in \mathbb{F}[\mathbf{z}]$. Because C computed in the for loop of lines 10-12 only maps some variables in $\mathbf{z}'$ to linear forms in $\mathbb{F}[\mathbf{z}]$, after this loop is executed, $\det(H_2)(BRA_0(C\mathbf{x} + \mathbf{b}) + \mathbf{d}) \in \mathbb{F}[\mathbf{z}]$. Thus, after A_0 is updated to be RA_0C and $\mathbf{b} := RA_0\mathbf{b}'$ in Step 13 of Procedure 2, $\det(H_2)(BA_0\mathbf{x} + B\mathbf{b} + \mathbf{d}) \in \mathbb{F}[\mathbf{z}]$. Since A_1' acts as identity on $\mathbf{z}$ and $A_1 = A_0A_1'$, $h' := \det(H_2)(BA_1\mathbf{x} + B\mathbf{b} + \mathbf{d}) \in \mathbb{F}[\mathbf{z}]$. Now, from the chain rule of derivatives we have that

$$\nabla h' = (BA_1)^T \left[\nabla \det(H_2)\right](BA_1\mathbf{x} + B\mathbf{b} + \mathbf{d}),$$

where $\nabla h'$ and $\nabla \det(H_2)$ are gradients of h' and $\det(H_2)$ with respect to $\mathbf{x}$, respectively. As h' does not contain any $\mathbf{u}$-variable,

$$\mathbf{0} = \left[\nabla h'\right]_{\mathbf{u}} = [(BA_1)^T]_{\mathbf{u}} \left[\nabla \det(H_2)\right](BA_1\mathbf{x} + B\mathbf{b} + \mathbf{d}),$$

where $[\nabla h']_{\mathbf{u}}$ is $\nabla h'$ restricted to entries corresponding to $\mathbf{u}$ and $[(BA_1)^T]_{\mathbf{u}}$ is $(BA_1)^T$ restricted to rows corresponding to $\mathbf{u}$. Thus,

$$[(BA_1)^T]_{\mathbf{u},\mathbf{x}'} \left[\nabla \det(H_2)\right]_{\mathbf{x}'} = -[(BA_1)^T]_{\mathbf{u},\mathbf{x}\setminus\mathbf{x}'} \left[\nabla \det(H_2)\right]_{\mathbf{x}\setminus\mathbf{x}'},$$

where $[(BA_1)^T]_{\mathbf{u},\mathbf{x}'}$ and $[(BA_1)^T]_{\mathbf{u},\mathbf{x}\setminus\mathbf{x}'}$ are the sub-matrices of $(BA_1)^T$ whose rows and columns are labelled by variables in $\mathbf{u}$, $\mathbf{x}'$ and $\mathbf{u}$, $\mathbf{x} \setminus \mathbf{x}'$ variables respectively. As $\mathscr{B} = [BA_1]_{\mathbf{x}',\mathbf{u}}\mathbf{u}$, $[(BA_1)^T]_{\mathbf{u},\mathbf{x}'}$ is invertible. By right multiplying its inverse on both sides of the above equation, we get that for all $x' \in \mathbf{x}'$, $\frac{\partial}{\partial x'} \det(H_2) \in \mathbb{F}\text{-span}\left\{\frac{\partial}{\partial x} \det(H_2) : x \in \text{var}(\det(H_2)) \setminus \mathbf{x}'\right\}$. Hence, $\mathbf{x}'$ is redundant for $\det(H_2)$. Then, as no variable $\mathbf{y} \setminus \mathbf{u}$ is present in $\det(H_2)$, the claim follows.

D.8 Proof of Claim 4.8

Pick any arbitrary $k \in \{s_1 + 1, \ldots, s_2\}$ and $y' \in \mathbf{y}_k$. If $\ell^{(2)}_{y'}$ contains a variable not in $\mathbf{z}_k \uplus \mathbf{y}_k$, because of Claim 4.5 and the fact that A_2' acts as identity on $\mathbf{z} \uplus (\mathbf{y} \setminus \mathbf{u})$, it must be in $\mathbf{z} \setminus \mathbf{z}_k$. Suppose that yy' is a term in the quadratic form along a skewed μ in T_k. Fix any $z \in \mathbf{z} \setminus \mathbf{z}_k$; if $z \in \text{var}\left(\ell^{(2)}_{y'}\right)$, then during the y-th iteration of the for loop of lines 2-8, g contains the monomial μz. We now argue that the only place in $f(A_2\mathbf{x} + \mathbf{b})$ which can contribute μz to g is $\mu\ell^{(2)}_{y'}$. This implies that the coefficient of μz in g, i.e., the coefficient of z in $\ell'_{y'}$ after Step 3 is equal to its coefficient in $\ell^{(2)}_{y'}$.

For μz to be present in g, μzy must be present in $f(A_2\mathbf{x} + \mathbf{b})$. We claim that μzy is not present in $\widehat{T}_{k'}(A_2\mathbf{x} + \mathbf{b})$ for any $k' \neq k$. If $k' \in [s_1]$, then this directly follows from Claim 4.3 and the fact that $A_1'A_2'$ act as identity on $\mathbf{z}_k$. For a $k' \in \{s_1 + 1, \ldots, s\}$, note that as $k \in \{s_1 + 1, \ldots, s_2\}$, $\deg(\mu) \geq 1$. Because y and variables in $\text{var}(\mu)$ are not in $\mathbf{z}_{k'} \uplus \mathbf{y}_{k'}$, y can only be present in $\ell^{(2)}_x$ for some $x \in \mathbf{x}_{k'}$ if x is a dangling variable along some skewed path. Hence any monomial of $\widehat{T}_{k'}(A_2\mathbf{x} + \mathbf{b})$ containing y can not contain any other variable in $\mathbf{z}_k \uplus \mathbf{y}_k$. So, μyz is not present in $\widehat{T}_{k'}(A_2\mathbf{x} + \mathbf{b})$.

Now, apart from $\ell^{(2)}_y$, y can only appear in $\ell^{(2)}_x$ for some $x \in \mathbf{x}_k$, if x is a dangling variable along some skewed path, say μ'. However, since $z \notin \mathbf{z}_k$, $z \notin \text{var}(\mu')$. So, μzy can not be present in $\mu'\ell^{(2)}_x$. This only leaves $\mu\ell^{(2)}_{y'}$ as the place that can contribute μz. Hence the coefficient of z in $\ell'_{y'}$ is equal to its coefficient in $\ell^{(2)}_{y_1}$. Now, notice that y' is not present in g in any iteration of the for

loop other than the y-th iteration. Hence, through out the execution of the loop, A'_3 only acts on y' during the y-th iteration. In this iteration, after Step 4 is executed, $\ell^{(2)}_{y'}(A'_3\mathbf{x}+\mathbf{b}')$ does not contain z as A'_3 is updated to map y' to $y'-\ell'_{y'}$. Since this is true for any $z \in \mathbf{z}\setminus\mathbf{z}_k$, the claim follows. □

D.9 Proof of Claim 4.9

Pick any arbitrary $k \in \{s_2+1,\ldots,s'\}$ and $y' \in \mathbf{y}_k$. If $\ell^{(2)}_{y'}$ contains a variable not in $\mathbf{y}_k$ (as $\mathbf{z}_k=\emptyset$), because of Claim 4.5 and the fact that A'_2 acts as identity on $\mathbf{z}\uplus(\mathbf{y}\setminus\mathbf{u})$, it must be in $\mathbf{z}$. Suppose that $T_k = yy'$. Fix any $z \in \mathbf{z}$; if $z \in \operatorname{var}\left(\ell^{(2)}_{y'}\right)$, then during the y-th iteration of the for loop of lines 2-8, g contains z. We now show that the only place in $f(A_2\mathbf{x}+\mathbf{b})$ that can contribute z to g is $\ell^{(2)}_{y'}$. Observe that for z to be present in g, yz must be present in $f(A_2\mathbf{x}+\mathbf{b})$. Apart from $\ell^{(2)}_{y}$, y is only present in $\ell^{(2)}_{x}$ if x is a dangling variable along a skewed path in some bad term $T_{k'}$ or $x=u_0$. However, Observation 4.5 implies that we can assume without loss of generality that the only place in $f(A_2\mathbf{x}+\mathbf{b})$ that contains zy is $\ell^{(2)}_{y}\ell^{(2)}_{y'}$. Thus, after Step 4 is executed, y' is mapped to an affine form in y' in $f(A_2A'_3\mathbf{x}+\mathbf{b})$. □

D.10 Proof of Claim 4.10

Fix a $k \in \{s_1+1,\ldots,s_2\}$ and an $x \in \mathbf{x}'_k$. Suppose that $y \in \mathbf{y}\setminus\mathbf{y}_k$ is present in $\ell^{(2)}_x$. Then, in the y-th iteration of the for loop of lines 2-8, the monomial μ' representing the skewed path corresponding to x is in $r(\mathbf{z})$. Observe that $\deg(\mu') \geq 1$. Note that the only time A'_3 translates the sole $\mathbf{u}$-variable u' in $\ell^{(2)}_x$ by a multiple of y is in the y-th iteration of the loop. So it suffices to prove that in this iteration, β in Step 6 is the coefficient of y in $\ell^{(2)}_x$. We do this by showing that the only place in $f(A_2\mathbf{x}+\mathbf{b})$ from which μ' can appear in g is from $\mu'\ell^{(2)}_x$.

For μ' to be present in g, $\mu'y$ must be present in $f(A_2\mathbf{x}+\mathbf{b})$. We claim that $\mu'y$ can not be present in $\widehat{T}_{k'}(A_2\mathbf{x}+\mathbf{b})$ for any $k' \neq k$. Because of Claims 4.8 and 4.9, variables in $\operatorname{var}(\mu')$ can only be present in $\ell^{(2)}_{x'}$ for some $x' \in \mathbf{x}_{k'}$ if x' is a dangling variable along some skewed path in $T_{k'}$ or $x'=u_0$. Hence any monomial of $\widehat{T}_{k'}(A_2\mathbf{x}+\mathbf{b})$ containing a variable in $\operatorname{var}(\mu')$ can not contain any other variable in $\mathbf{z}_k\uplus\mathbf{y}_k$. So, $\mu'y$ is not present in $\widehat{T}_{k'}(A_2\mathbf{x}+\mathbf{b})$.

Now, in $\widehat{T}_k(A_2\mathbf{x}+\mathbf{b})$, y is only present in $\ell^{(2)}_{x'}$ for some $x' \in \mathbf{x}_k$ if x' is a dangling variable along some skewed path. However, if that skewed path is μ'', then the monomial present in $\widehat{T}_k(A_2\mathbf{x}+\mathbf{b})$ is $\mu''y$. Hence $\mu''\ell^{(2)}_{x'}$ can contain the monomial $\mu'y$ only if $\mu''=\mu'$ and $x'=x$. Thus only $\mu'\ell^{(2)}_x$ contributes μ' to g, and β is precisely the coefficient of y in $\ell^{(2)}_x$. Hence, after A'_3 has been updated to map u' to $u'-\beta y$ in Step 6, $\ell^{(2)}_x(A'_3\mathbf{x})$ does not contain y.

For any $y \in \ell^{(2)}_{u_0}$, in the y-th iteration of the loop, $r(\mathbf{z})$ contains a constant, say β. Because u_0 is only translated by a constant multiple of y in the y-th iteration of the loop, it is sufficient to show that β is the coefficient of y in $\ell^{(2)}_{u_0}$. If $y \in \mathbf{y}_k$ for some $k \in \{s_1+1,\ldots,s_2\}$, then every monomial in $f(A_2\mathbf{x}+\mathbf{b})$ containing y must also contain a skewed path. Observation 4.6 implies that A'_3 maps the top quadratic from to $\sum_{k=s_2+1}^{s'} y_{k,1}y_{k,2}$. Thus βy is also not present in the top quadratic form. Hence β is exactly the coefficient of y in $\ell^{(2)}_{u_0}$. Also, in this case when Step 6 is executed, $u'=u_0$. So after A'_3 has been updated to map u' to $u'-\beta y$, $\ell^{(2)}_{u_0}(A'_3\mathbf{x})$ does not contain y. □

D.11 Proof of Claim 4.11

Fix any $k \in \{s_1+1, \ldots, s_2\}$ and $x \in \mathbf{x}'_k$ which is not a bad dangling variable. If the corresponding skewed path is μ, then $\deg(\mu) \geq 2$. Also, from Observation 4.3, if the sole $\mathbf{u}$ variable in $\ell_u^{(3)}$ is u, then $(\mu, u) \in V$. We analyse the iteration of the for loop of lines 11-16 corresponding to (μ, u). Fix any $z \in \mathbf{z} \setminus \mathbf{z}_k$. We show that in the z-th iteration of the for loop of lines 12-15 the coefficient of μz in g is exactly the coefficient of z in $\ell_x^{(3)}$.

As $\deg(\mu) \geq 2$, μz is not present in $\widehat{T}_{k'}(A_3\mathbf{x} + \mathbf{b})$ for any $k' \neq k$. Also, in $\widehat{T}_k(A_3\mathbf{x} + \mathbf{b})$, z is only present in $\ell_x^{(3)}$ for some $x \in \mathbf{x}_k$, if x is a dangling variable along a skewed path. However, if the skewed path corresponding to x is μ', then we get the monomial $\mu' z$ form $\mu' \ell_x^{(3)}$. This means that $\mu' = \mu$ and hence in $\widehat{T}_k(A_3\mathbf{x} + \mathbf{b})$, μz is only obtained from $\mu \ell_x^{(3)}$. This means that in Step 14, α is precisely the coefficient of z in $\ell_x^{(3)}$. Hence, after that step is executed and A'_4 updated to map u to $u - \alpha z$, $\ell_x^{(3)}(A'_4\mathbf{x})$ does not contain z. Also, observe that the only monomials containing z in $\widehat{T}_{k'}(A_3(\mathrm{var}(\mu'),\ z,\ \mathbf{x} \setminus (\mathrm{var}(\mu') \uplus \{z\}) = \mathbf{0}) + \mathbf{b})$, for $k' \neq k$ can be of degree at most 2. Further any monomial containing z in $\widehat{T}_k(A_3(\mathrm{var}(\mu'),\ z,\ \mathbf{x} \setminus (\mathrm{var}(\mu') \uplus \{z\}) = \mathbf{0}) + \mathbf{b})$ must look like $\mu' z$, where μ' is a sub-monomial of μ. Hence, $\frac{\partial g}{\partial z}$ is sparse and can be interpolated efficiently. □

D.12 Proof of Claim 4.12

We prove the claim by showing that the following loop invariant holds: The matrix A'_4 computed after the i-th iteration of the loop is such that for all $j \leq i$, if $x_j \in \mathbf{x}'_k$, then $\widehat{\ell}_{x_j}^{(3)}(A'_4\mathbf{x}) \in \mathbb{F}[\mathbf{z}_k \uplus \mathbf{y}_k]$. Suppose that the invariant is true before the execution of the i-th iteration of the loop; it is trivially true before the first iteration. Suppose that $x_i \in \mathbf{x}'_k$. First we consider the z-th iteration of the for loop of lines 18-21 for a $z \notin \mathbf{z}_k \cup \{z_1, \ldots, z_m\}$. Since for any $k \in [s]$, the only $x \in \mathbf{x}_k$ such that $\ell_x^{(3)}(A'_4\mathbf{x})$ contains variables not in $\mathbf{z}_k$ are $x \in \{u_0, x_1, \ldots, x_m\}$, the only place in $f(A_3A'_4\mathbf{x} + \mathbf{b})$ that can contribute $z_i z$ to g is $z_i \cdot \ell_{x_i}^{(3)}(A'_4\mathbf{x})$. Hence, in Step 20 α is exactly the coefficient of z in $\ell_{u_i}^{(3)}(A'_4\mathbf{x})$. Thus after that step is executed and A'_4 is updated to map the sole $\mathbf{u}$ variable u_i in $\ell_{x_i}^{(3)}(A'_4\mathbf{x})$ to $u_i - \alpha z$, $\ell_{x_i}^{(3)}(A'_4\mathbf{x})$ does not contain z. For a $z \in \{z_1, \ldots, z_{i-i}\}$, the assumption that the loop invariant is true before the i-th iteration and Observation 4.7 imply that $z_i z$ is not a monomial in g and hence z is not present in $\frac{\partial g}{\partial z}$. On the other hand, for all $z \in \{z_{i+1}, \ldots, z_m\}$, Observation 4.7 implies that in Step 20 α is exactly the coefficient of z in $\ell_{x_i}^{(3)}(A'_4\mathbf{x})$. Hence, after that step is executed and A'_4 updated to map the sole $\mathbf{u}$ variable u_i in $\ell_{x_i}^{(3)}(A'_4\mathbf{x})$ to $u_i - \alpha z$, $\ell_{x_i}^{(3)}(A'_4\mathbf{x})$ does not contain z. Notice that this also implies that the monomial $z_i z$ is no longer present in $f(A'_4\mathbf{x} + \mathbf{b})$. Also, in the i-th iteration, A'_4 does not act on any variable other than u_i. Hence, the invariant is also true after the execution of this iteration. □

D.13 Proof of Claim 4.13

There are two cases.

Case 1: $k \in [s_1]$, say $T_k = Q_{k,1} \cdots Q_{k,m_k}$, $m_k \geq 2$ or neither $Q_{k,1}$ nor $Q_{k,2}$ is linear, $\widehat{T}_k = \widehat{Q}_{k,1} \cdots \widehat{Q}_{k,m_k}$, and $\widehat{Q}_{k,l} = Q_{k,l}(B\mathbf{x} + \mathbf{d})$ for all $l \in [m_k]$. Then from Claim 3.1, $\widehat{Q}_{k,1}(RA_0\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(RA_0\mathbf{x})$ are irreducible factors of $h(A_0\mathbf{x})$ where R, A_0, and h are as just before Step 8 of Procedure 2 is executed.

Because $\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_m$ are returned by Make-Factors-Var-Disjoint($h(A_0\mathbf{x})$), from Claim 2.4, for every $l \in [m_k]$, there exists an $i \in [m]$ such that var $\left(\widehat{Q}_{k,l}(RA_0\mathbf{x})\right) \subseteq \widehat{\mathbf{z}}_i$, where A_0 is as after Step 8 of Procedure 2 has been executed. It follows from Observation 2.2 that every variable in var$(Q_{k,l})$ is mapped to a linear from in $\widehat{\mathbf{z}}_i$ by BRA_0. As C only maps some of these linear forms to constant multiples of variables in $\widehat{\mathbf{z}}_i$, even after A_0 is updated to be RA_0C in Step 13 of Procedure 2, var $\left(\widehat{Q}_{k,l}(A_0\mathbf{x}+\mathbf{b})\right) =$ var $\left(\widehat{Q}_{k,l}(A_0\mathbf{x})\right) \subseteq \widehat{\mathbf{z}}_i$. Because in Procedures 3 and 4, $A'_1, \ldots, A'_4$ act as identity on $\mathbf{z}$, var $\left(\widehat{Q}_{k,l}(A\mathbf{x}+\mathbf{b})\right) \subseteq \widehat{\mathbf{z}}_i$ where A and $\mathbf{b}$ are as after Step 3 of Algorithm 1.

If var $\left(\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b})\right) \cup \cdots \cup$ var $\left(\widehat{Q}_{k,m_k}(A\mathbf{x}+\mathbf{b})\right) \subseteq \widehat{\mathbf{z}}_i$, then var $\left(\widehat{T}_k(A\mathbf{x}+\mathbf{b})\right)$ is clearly contained in a single connected component of G. So suppose that there exist disjoint sets $I_1, I_2 \subseteq [m_k]$ and $i \neq j \in [m]$ such that $\cup_{l\in I_1}$var $\left(\widehat{Q}_{k,l}(A\mathbf{x}+\mathbf{b})\right) \subseteq \widehat{\mathbf{z}}_i$, $\cup_{l\in I_2}$var $\left(\widehat{Q}_{k,l}(A\mathbf{x}+\mathbf{b})\right) \subseteq \widehat{\mathbf{z}}_j$, and $\cup_{l\notin I_1\uplus I_2}$var $\left(\widehat{Q}_{k,l}(A\mathbf{x}+\mathbf{b})\right) \cap (\widehat{\mathbf{z}}_i \uplus \widehat{\mathbf{z}}_j) = \emptyset$. Then, for any $z_1 \in \widehat{\mathbf{z}}_i$ and $z_2 \in \widehat{\mathbf{z}}_j$, $\frac{\partial^2 f(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial z_2} = \frac{\partial^2 \widehat{T}_k(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial z_2} = \frac{\partial}{\partial z_1}\left(\prod_{l\in I_1} \widehat{Q}_{k,l}(A\mathbf{x}+\mathbf{b})\right) \cdot \frac{\partial}{\partial z_2}\left(\prod_{l\in I_2} \widehat{Q}_{k,l}(A\mathbf{x}+\mathbf{b})\right) \cdot \left(\prod_{l\notin I_1\uplus I_2} \widehat{Q}_{k,l}(A\mathbf{x}+\mathbf{b})\right) \neq 0$. So the edge $\{\widehat{\mathbf{z}}_i, \widehat{\mathbf{z}}_j\}$ is added to G, and var $\left(\widehat{T}_k(A\mathbf{x}+\mathbf{b})\right)$ is in a single connected component of G.

Case 2: $k \in \{s_1+1, \ldots, s_2\}$, say $\widehat{T}_k = \widehat{Q}_{k,1}\widehat{Q}_{k,2}$, where $\widehat{Q}_{k,1}$ is in the orbit of a variable. If there exists an i such that var $\left(\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b})\right) \cup$ var $\left(\widehat{Q}_{k,2}(A\mathbf{x}+\mathbf{b})\right) \subseteq \widehat{\mathbf{z}}_i$, then var $\left(\widehat{T}_k(A\mathbf{x}+\mathbf{b})\right)$ is clearly contained in a single connected component of G. Otherwise, it follows from Lemma 4.3 that $\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b})$ is a constant multiple of a variable, say $z_1 \in \widehat{\mathbf{z}}_i$. Let x be any variable in $\widehat{Q}_{k,2}(A\mathbf{x}+\mathbf{b})$. First suppose that $x \in \widehat{\mathbf{z}}_j$ for some $j \neq i$ and $x = z_2$. Then $\frac{\partial^2 f(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial z_2} = \frac{\partial^2 \widehat{T}_k(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial z_2} = \frac{\partial}{\partial z_1}\left(\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}) \cdot \frac{\partial \widehat{Q}_{k,2}(A\mathbf{x}+\mathbf{b})}{\partial z_2}\right)$. As, $\frac{\partial \widehat{Q}_{k,2}(A\mathbf{x}+\mathbf{b})}{\partial z_2} \neq 0$, $z_1 \in$ var $\left(\widehat{Q}_{k,1}(A\mathbf{x}+\mathbf{b}) \cdot \frac{\partial \widehat{Q}_{k,2}(A\mathbf{x}+\mathbf{b})}{\partial z_2}\right)$. Thus, $\frac{\partial^2 f(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial z_2} \neq 0$ and the edge $\{\widehat{\mathbf{z}}_i, \widehat{\mathbf{z}}_j\}$ is added to G. Now, if $x \in \mathbf{y}$ and $x = y$, even then using the same argument as above, $\frac{\partial^2 f(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial y} \neq 0$ and the edge $\{\widehat{z}, y\}$ is added to G. Thus var $\left(\widehat{T}_k(A\mathbf{x}+\mathbf{b})\right)$ is contained in a single connected component of G.

So for all $k \in [s_2]$, var $\left(\widehat{T}_k(A\mathbf{x}+\mathbf{b})\right)$ is contained in a single connected component of G. Also, for $k \neq k' \in [s']$ and any $z_1 \in \widehat{T}_k(A\mathbf{x}+\mathbf{b})$, $z_2 \in \widehat{T}_{k'}(A\mathbf{x}+\mathbf{b})$, $\frac{\partial^2 f(A\mathbf{x}+\mathbf{b})}{\partial z_1 \partial z_2} = 0$. Thus, var $\left(\widehat{T}_k(A\mathbf{x}+\mathbf{b})\right)$ corresponds to a connected component in G. Further for any $y_1, y_2 \in$ var $\left(\widehat{T}_{s_2+1}(A\mathbf{x}+\mathbf{b})\right) \uplus \ldots \uplus$ var $\left(\widehat{T}_{s'}(A\mathbf{x}+\mathbf{b})\right)$ observe that $\frac{\partial^2 f(A\mathbf{x})}{\partial y_1 \partial y_2}$ is never computed, hence they are in distinct connected components of G of size 1 each. u_0 is also in a connected component containing just itself. Hence the only connected components of G with more then 1 variable correspond to var $\left(\widehat{T}_1(A\mathbf{x}+\mathbf{b})\right), \ldots,$ var $\left(\widehat{T}_{s_2}(A\mathbf{x}+\mathbf{b})\right)$. □

D.14 Proof of Claim 4.15

Fix an $i \in [|\mathbf{z}_k|]$. As $\mathbf{a}_i$ is chosen randomly, $\deg(\widehat{Q}(t\mathbf{a}_i)) \geq 2$ with high probability. Notice that there exist $\widehat{Q}'_i, \beta_{i,0}, \beta_{i,1}$ such that $\widehat{Q}(t\mathbf{a}_i) \cdot \widehat{Q}'_i(t) + \beta_{i,1} \cdot t + \beta_{i,0} = \widehat{T}(t\mathbf{a}_i)$ is satisfied; one solution is $\widehat{Q}'_i = \left(\widehat{T}_k(A\mathbf{x}+\mathbf{b})/\widehat{Q}\right)(t\mathbf{a}_i)$, $\beta_{i,1} = (\sum_{z\in\mathbf{z}_k} c_z \cdot z)(\mathbf{a}_i)$, and $\beta_{i,0} = \gamma'$. We claim that this solution is unique

with high probability. Suppose that there existed two solutions $\widehat{Q}'_i, \beta_{i,0}, \beta_{i,1}$ and $\widehat{Q}''_i, \beta'_{i,0}, \beta'_{i,1}$. Then $\widehat{Q}(t\mathbf{a}_i) \cdot (\widehat{Q}'_i(t) - \widehat{Q}''_i(t)) = (\beta'_{i,1} - \beta_{i,1}) \cdot t + \beta'_{i,0} - \beta_{i,0}$. As $\deg(\widehat{Q}(t\mathbf{a}_i)) \geq 2$ with high probability, this is only possible if $\widehat{Q}'_i(t) = \widehat{Q}''_i(t)$, $\beta_{i,1} = \beta'_{i,1}$ and $\beta_{i,0} = \beta'_{i,1}$. In particular $\beta_{i,1} = (\sum_{z \in \mathbf{z}_k} c_z \cdot z)(\mathbf{a}_i)$ and $\beta_{i,0} = \gamma'$. Thus, after we interpolate $\sum_{z \in \mathbf{z}_k} \alpha_z z$ using $\beta_{i,1}, \ldots, \beta_{|\mathbf{z}_k|,1}$ and set $\ell_k = \sum_{z \in \mathbf{z}_k} \alpha_z z + \beta_{1,0}$, $\widehat{T} = \widehat{T}_k(A\mathbf{x} + \mathbf{b}) + [\ell(A\mathbf{x} + \mathbf{b})]_{\mathbf{z}_k} + \gamma' - \ell_k = \widehat{T}_k(A\mathbf{x} + \mathbf{b})$ is reducible. □

D.15 Proof of Claim 4.16

Fix an $i \in [|\mathbf{z}_k| - 1]$. Because $\deg(T_k) \geq 3$, Lemma 4.3 implies that $\widehat{Q}$ is a variable. Notice that there exist $\widehat{Q}'_i(z,t), \beta_{i,2}, \beta_{i,1}, \beta_{i,0}$ satisfying $z \cdot \widehat{Q}'_i(z,t) + \beta_{i,2} \cdot z + \beta_{i,1} \cdot t + \beta_{i,0} = \widehat{T}(z, \mathbf{z}_k \setminus \{z\} = \mathbf{a}_i \cdot t)$; one such solution is $\widehat{Q}'_i = \left(\widehat{T}_k(A\mathbf{x} + \mathbf{b})/\widehat{Q}\right)(z, t\mathbf{a}_i)$, $\beta_{i,2} = c_z$, $\beta_{i,1} = (\sum_{z' \in \mathbf{z}_k \setminus \{z\}} c_{z'} \cdot z')(\mathbf{a}_i)$ and $\beta_{i,0} = \gamma'$. We now show that $\beta_{i,1}$, and $\beta_{i,0}$ are unique with high probability. Suppose there are two solutions $\widehat{Q}'_i(z,t), \beta_{i,2}, \beta_{i,1}, \beta_{i,0}$ and $\widehat{Q}''_i(z,t), \beta'_{i,2}, \beta'_{i,1}, \beta'_{i,0}$. Then, $z \cdot (\widehat{Q}'_i(z,t) - \widehat{Q}''_i(z,t)) + (\beta_{i,2} - \beta'_{i,2}) \cdot z = (\beta'_{i,1} - \beta_{i,1}) \cdot t + (\beta'_{i,0} - \beta_{i,0})$. By putting $z = 0$ it can be seen that $\beta_{i,1} = \beta'_{i,1} = (\sum_{z' \in \mathbf{z}_k \setminus \{z\}} c_{z'} \cdot z')(\mathbf{a}_i)$ and $\beta_{i,0} = \beta'_{i,0} = \gamma'$. Thus, after we interpolate $\sum_{z' \in \mathbf{z}_k \setminus \{z\}} \alpha_{z'} z'$ using $\beta_{i,1}, \ldots, \beta_{|\mathbf{z}_k|,1}$ and set $\ell_k = \sum_{z' \in \mathbf{z}_k \setminus \{z\}} \alpha_{z'} z' + \beta_{1,0}$, $\widehat{T} = \widehat{T}_k(A\mathbf{x} + \mathbf{b}) + [\ell(A\mathbf{x} + \mathbf{b})]_{\mathbf{z}_k} + \gamma' - \ell_k = \widehat{T}_k(A\mathbf{x} + \mathbf{b})$ is reducible. □

D.16 Proof of Claim 4.17

One direction is simple. If $\ell = c \cdot \widehat{Q}_l$ for some $l \in [m]$, then it is clear that T is reducible.

For the other direction, pick a $z \in \mathrm{var}(\ell)$ and let its coefficient be c. Note that there must exist an $l \in [m]$ such that $z \in \mathrm{var}(Q_l)$, for otherwise T is in the orbit of the $+$-rooted ROF $Q_1 \cdots Q_m + z$ and therefore irreducible. So assume without loss of generality that $z \in \mathrm{var}(Q_1)$. Let $T = r_1 \cdot r_2$ where r_1 is an irreducible factor of T containing z and r_2 is not necessarily irreducible. Note that as T is multilinear r_1 and r_2 are variable disjoint. Now,

$$r_1 r_2 = Q_1 Q + c \cdot z + \ell'$$

where $Q = Q_2 \cdots Q_m$ and $\ell = c \cdot z + \ell'$. Let $Q_1 = g_1 \cdot z + g_2$ and $r_1 = w_1 \cdot z + w_2$ where g_1, g_2, w_1, and w_2 are z-free polynomials. Then,

$$(w_1 \cdot z + w_2) \cdot r_2 = (g_1 \cdot z + g_2) \cdot Q + c \cdot z + \ell'. \tag{10}$$

Observation D.1. $w_1 \in \mathbb{F}^\times$.

Proof. Taking derivatives with respect to z on both sides of Equation (10), we see that $w_1 \cdot r_2 = g_1 \cdot Q + c$. If g_1 is a non-constant polynomial, then the latter is a $+$ rooted ROF and therefore from Fact 2.5, irreducible. Hence, r_2 is irreducible and $g_1 \in \mathbb{F}$.

If g_1 is a constant and g_2 is also a constant, then as $Q_1 Q + u$ is a canonical ROF, either Q must be a product of at least two $+$ rooted ROFs or if it is just a single $+$ rooted ROF, then it can not have a constant attached to its top-most $+$ gate. In either of these cases $g_1 \cdot Q + c$ is irreducible and hence we again get that r_2 is irreducible and $g_1 \in \mathbb{F}$.

Suppose that g_1 is a constant, but g_2 is not. As Q is non-constant (since $m \geq 2$), there exists a $z_2 \in \mathrm{var}(Q)$. Then, for any $z_1 \in \mathrm{var}(g_2)$, as zz_1 does not appear on the right side of Equation (10), $z_1 \notin \mathrm{var}(w_1)$ or $\mathrm{var}(r_2)$. Also, for any $z'_2 \in \mathrm{var}(Q)$, $z'_2 \notin \mathrm{var}(w_1)$. This is so because, for any $z'_1 \in \mathrm{var}(g_2)$, $z'_1 z'_2$ appears on the right side of Equation (10). If z'_1 were to be in $\mathrm{var}(w_1)$, then $z'_1 z'_2$

can not appear on the left side of (10) since $z_1' \notin \text{var}(w_1)$ or $\text{var}(r_2)$. Thus, no variable in $\text{var}(g_2)$ or $\text{var}(Q)$ can be present in $\text{var}(w_1)$ forcing w_1 to be a constant. □

Assume without loss of generality that $w_1 = 1$. Thus,

$$(z + w_2) \cdot r_2 = (g_1 \cdot z + g_2) \cdot Q + c \cdot z + \ell',$$

which implies that $r_2 = g_1 \cdot Q + c$. Hence,

$$(z + w_2) \cdot (g_1 \cdot Q + c) = (g_1 \cdot z + g_2) \cdot Q + c \cdot z + \ell'.$$

Observation D.2. $g_1 \in \mathbb{F} \setminus \{0\}$.

Proof. By contradiction. Suppose that g_1 is a non-constant polynomial. Then we have

$$\begin{aligned} &(z + w_2) \cdot (g_1 \cdot Q + c) = (g_1 \cdot z + g_2) \cdot Q + c \cdot z + \ell' \\ \implies &(w_2 \cdot g_1 - g_2)Q = -c \cdot w_2 + \ell'. \end{aligned}$$

Suppose that $w_2 \cdot g_1 - g_2 \neq 0$. Then Q and w_2 must be variable disjoint, we get

1. Q is linear and so as it is a canonical ROF is an affine form in a single variable,
2. $w_2 \cdot g_1 - g_2 \in \mathbb{F}^\times$, say it is c', and
3. w_2 is an affine form.

Hence, $Q_1 = g_1 \cdot z + g_2 = g_1 \cdot z + g_1 \cdot w_2 - c'$. We claim that c' must be 0. It given that $Q_1 Q + u$ is a canonical ROF and Q is an affine form in a single variable. Thus it follows from property 6 of the definition of a canonical ROF that Q_1 can not have a constant attached to its top-most $+$ gate; hence $c' = 0$. However this contradicts our assumption that $c' = w_2 \cdot g_1 - g_2 \neq 0$. Hence, $w_2 \cdot g_1 - g_2 = 0$. So, $g_2 = w_2 \cdot w_1$. This implies that $Q_1 = g_1(z + w_2)$, which implies that $g_1 \in \mathbb{F}^\times$ because Q_1 being a $+$-rooted ROF is irreducible. □

Assume without loss of generality that $g_1 = 1$; if it is not, then replace Q by $g_1 \cdot Q$ and g_2 by $g_1^{-1} \cdot g_2$. Then,

$$\begin{aligned} &(z + w_2)(Q + c) = (z + g_2) \cdot Q + c \cdot z + \ell' \\ \implies &(w_2 - g_2)Q = -c \cdot w_2 + \ell'. \end{aligned}$$

Then, just as before, Q and w_2 are variable disjoint. Thus,

1. Q is linear and so as it is a canonical ROF is an affine form in a single variable,
2. $w_2 - g_2 \in \mathbb{F}^\times$, say it is c'', and
3. w_2 is an affine form.

Suppose $w_2 - g_2 \neq 0$. Hence, $Q_1 = z + w_2 - c''$ and as Q_1 is canonical, w_2 has to be a constant. But then, Q_1 and Q are both affine which contradicts the hypothesis that $Q_1 \cdots Q_m$ is not a quadratic polynomial. Thus, $w_2 = g_2$. Then,

$$\begin{aligned} &(z + w_2)(Q + c) = (z + w_2) \cdot Q + \ell \\ \implies &(z + w_2) \cdot c = \ell. \end{aligned}$$

Now, as $Q_1 = (z + w_2)$ is a canonical ROF, $w_2 \in \mathbb{F}$. Hence, $\ell = c \cdot Q_1$. □

D.17 Proof of Lemma 4.5

As $f(A\mathbf{x}) = \widehat{T}_1(A\mathbf{x}) + \cdots + \widehat{T}_s(A\mathbf{x}) + \gamma$, $\widehat{T}_1(A\mathbf{x}), \ldots, \widehat{T}_s(A\mathbf{x})$ are variable disjoint, and for all $k \in [s_2]$ $\mathbf{z}_k = \text{var}\left(\widehat{T}_k(A\mathbf{x})\right)$, $f\left(A\left(\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k = \mathbf{0}\right)\right) = \widehat{T}_k(A\mathbf{x}) + \gamma'$ for some $\gamma' \in \mathbb{F}$. So all that needs to be done to obtain black-box access to $\widehat{T}_k(A\mathbf{x})$ is to find γ' and subtract it from $g = f\left(A\left(\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k = \mathbf{0}\right)\right)$. We show that this is exactly what the procedure does.

In the procedure, N is the set of irreducible factors of $\det\left(H_g\right)$. Suppose that $\widehat{T}_k = \widehat{Q}_{k,1} \cdots \widehat{Q}_{k,m_k}$. It follows from Claim 3.1 and Fact 2.8, that a non-zero constant multiple of at least one of the factors $\widehat{Q}_{k,1}(A\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(A\mathbf{x})$ is in N along with some other 'bad' factors. First let us analyse the behaviour of the for loop of lines 3-7 when r is a constant multiple of one of the $\widehat{Q}_{k,1}(A\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(A\mathbf{x})$. In this case, there exist $r'(t)$ and $\beta \in \mathbb{F}$ such that $r(t\mathbf{a})r'(t) + \beta = g(t\mathbf{a})$; one solution is $r'(t) = \left(\widehat{T}_k(A\mathbf{x})/r\right)(A(t\mathbf{a}))$ and $\beta = \gamma'$. $r'(t)$ and β can be discovered as follows: first interpolate $r(t\mathbf{a})$ and $g(t\mathbf{a})$ which are univariate polynomials in t. Treat the coefficients of $r'(t)$ and β as unknowns. Then, by equating the coefficients of monomials on both sides of $r(t\mathbf{a})r'(t) + \beta = g(t\mathbf{a})$, we get a system of linear equations in these unknowns which the procedure solves. As mentioned before, this system has a solution, we now show it is unique.

Suppose that there existed two solutions $r'_1(t), \beta_1$ and $r'_2(t), \beta_2$. Then, $r(t\mathbf{a})\left(r'_1(t) - r'_2(t)\right) = \beta_2 - \beta_1$. Because $\mathbf{a}$ is chosen randomly, with high probability $r(t\mathbf{a})$ is a non-constant polynomial in t. Thus, this is only possible when $r'_1(t) = r'_2(t)$ and $\beta_1 = \beta_2 = \gamma'$. Hence, if r is a constant multiple of one of the $\widehat{Q}_{k,1}(A\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(A\mathbf{x})$, then $\beta = \gamma'$ and $g - \beta$ is reducible. Thus, the procedure returns a black-box of $g - \beta = \widehat{T}_k(A\mathbf{x})$.

On the other hand, when r is one of the bad factors, there are two cases: $\beta = \gamma'$ and $\beta \neq \gamma'$. In the first case there is noting to prove. On the other hand, if $\beta \neq \gamma'$, then notice that $g + \gamma' - \beta$ is in the orbit of $T_k + \gamma' - \beta$. As the latter is a $+$-rooted ROF, Fact 2.5 implies that it is irreducible. Hence $g - \beta$ is also irreducible. Now, the fact that the for loop was executed for this bad factor implies that in all of the previous iterations, r must have been a bad factor, for otherwise as seen above, the loop would have terminated. This along with the fact that N contains a constant multiple of at least one of the $\widehat{Q}_{k,1}(A\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(A\mathbf{x})$ implies that in this case, the next iteration of the loop will be executed. This will continue to happen until the loop is executed for an r which is a non-zero constant multiple of one of the $\widehat{Q}_{k,1}(A\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(A\mathbf{x})$ and γ' is discovered.

Notice that $\widehat{T}_k(A\mathbf{x}) = f\left(A\left(\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k = \mathbf{0}\right)\right) - \beta$. Hence, to query $\widehat{T}_k(A\mathbf{x})$ at $\mathbf{z}_k = \mathbf{a}$, for some $\mathbf{a} \in \mathbb{F}^{|\mathbf{z}_k|}$, $f(A\mathbf{x})$ just needs to be queried at the point $(\mathbf{z}_k = \mathbf{a}, \mathbf{x} \setminus \mathbf{z}_k = \mathbf{0})$ and then β, which is a fixed, known constant, subtracted from the result. Now as A is known to us, to query $f(A\mathbf{x})$ at any point, we just need to query f at one point. □

D.18 Proof of Claim 4.18

Fix a $k \in [s_2]$. The hypothesis of Claim 4.5 is satisfied. Hence after Step 16 is executed, $\widehat{T}$ is the black-box of $\widehat{T}_k(A\mathbf{x})$. Suppose that $\widehat{T}_k(A\mathbf{x}) = \widehat{Q}_{k,1}(A\mathbf{x}) \cdots \widehat{Q}_{k,m_k}(A\mathbf{x})$, the corresponding term $T'_k(P_0\mathbf{x})$ of $\mathsf{C}'(P_0\mathbf{x})$ is a product of $+$-rooted canonical sub-ROFs $Q_{k,1}, \ldots, Q_{k,m_k}$ and for all $l \in [m_k]$, $\widehat{Q}_{k,l}(A\mathbf{x}) = Q_{k,l}(B'\mathbf{x} + \mathbf{d}')$. The factors $\widehat{Q}_1, \ldots, \widehat{Q}_{m_k}$ of $\widehat{T}$ computed in Step 17 are non-zero constant multiples of $\widehat{Q}_{k,1}(A\mathbf{x}), \ldots, \widehat{Q}_{k,m_k}(A\mathbf{x})$, respectively, say they are $c_1\widehat{Q}_{k,1}(A\mathbf{x}), \ldots, c_{m_k}\widehat{Q}_{k,m_k}(A\mathbf{x})$; here $\prod_{l\in[m_k]} c_l = 1$. Now, as $Q_{k,1}, \ldots, Q_{k,m_k}$ are variable disjoint ROFs, $N_{ess}\left(Q_{k,1} \cdots Q_{k,m_k}\right) = N_{ess}\left(Q_{k,1}\right) + \cdots + N_{ess}\left(Q_{m_k}\right)$. Also, for all $l \in [m_k]$ $N_{ess}\left(\widehat{Q}_l\right) = N_{ess}\left(\widehat{Q}_{k,l}(A\mathbf{x})\right) = N_{ess}\left(Q_{k,l}\right)$

and similarly $N_{ess}\left(\widehat{Q}_1 \cdots \widehat{Q}_{m_k}\right) = N_{ess}\left(Q_{k,1} \cdots Q_{k,m_k}\right)$. This means that $N_{ess}\left(\widehat{Q}_1 \cdots \widehat{Q}_{m_k}\right) = N_{ess}\left(\widehat{Q}_1\right) + \cdots + N_{ess}\left(\widehat{Q}_{m_k}\right)$. So from Claim 2.3, there exists an $A_{k,0} \in \mathrm{GL}(|\mathbf{z}_k|, \mathbb{F})$ such that $\widehat{Q}_1(A_{k,0}\mathbf{z}_k), \ldots, \widehat{Q}_{m_k}(A_{k,0}\mathbf{z}_k)$ are variable disjoint. It also implies that $\widehat{Q}_1(A_{k,0}\mathbf{z}_k), \ldots, \widehat{Q}_{m_k}(A_{k,0}\mathbf{z}_k)$ do not contain any redundant variables. In Step 18, $\widehat{Q}_l$ has been updated to be $\widehat{Q}_l(A_{k,0}\mathbf{z}_k)$ for all $l \in [m_k]$. Now $\widehat{Q}_l(A_{k,0}\mathbf{z}_k)$, $|\mathrm{var}(Q_{k,l})| = |\mathbf{z}_{k,l}|$, where $\mathbf{z}_{k,l} = \mathrm{var}\left(\widehat{Q}_l(\mathbf{z}_k)\right)$. So, there exists a permutation matrix $P_{k,0} \in M(|\mathbf{z}_k|, \mathbb{F})$ such that for all $l \in [m_k]$, $\mathrm{var}(Q_{k,l}(P_{k,0}\mathbf{z}_k)) = \mathbf{z}_{k,l}$.

Much like the outer loop, the l-th iteration of the inner loop of lines 20-23, also only works with $\widehat{Q}_l(\mathbf{z}_k)$ and $\mathbf{z}_{k,l}$; so we can also look at an iteration of this loop in isolation. We now analyse the l-th iteration of this loop for some $l \in [m_k]$. $\mathbf{a}$ is a random vector of size $|\mathbf{z}_k|$ and $\mathbf{a}'$ is $\mathbf{a}$ restricted to entries corresponding to $\mathbf{z}_k \setminus \mathbf{z}_{k,l}$. Also $\beta_l = \prod_{l' \in [m_k] \setminus \{l\}} \widehat{Q}_{l'}(\mathbf{z}_{k,l}, \mathbf{z}_k \setminus \mathbf{z}_{k,l} = \mathbf{a}')$. Because $\mathbf{a}$ is random, $\beta_l \neq 0$ with high probability. Thus $\widehat{Q}_l = \beta_l^{-1}\widehat{T}\left(A_{k,0}\left(\mathbf{z}_{k,l}, \mathbf{z}_k \setminus \mathbf{z}_{k,l} = \mathbf{a}'\right)\right) = c_l \widehat{Q}_{k,l}(A(A_{k,0}\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k))$. Now consider a product-depth Δ canonical ROF Q_l obtained by multiplying $Q_{k,l}(P_{k,0}\mathbf{z}_k)$ by c_l, pushing it down to the leaves, and removing it from any non-constant leaf. Let $B'' = {P'_{k,0}}^{-1} B' A'_{k,0}$, where $P'_{k,0} \in M(n, \mathbb{F})$ maps every $z \in \mathbf{z}_k$ to $P_{k,0} \circ z$ and every other variable to itself, while $A'_{k,0} \in \mathrm{GL}(n, \mathbb{F})$ maps every $z \in \mathbf{z}_k$ to $A_{k,0} \circ z$ and every other variable to itself. Also, let $\mathbf{d}'' = {P'_{k,0}}^{-1}\mathbf{d}'$. It can be verified that $\widehat{Q}_l = Q_l\left(B''\mathbf{x} + \mathbf{d}''\right)$. To recursively perform equivalence test on $\widehat{Q}_l$ we shall show that there exists a $B_l \in \mathrm{GL}(|\mathbf{z}_{k,l}|, \mathbb{F})$ and a $\mathbf{d}_l \in \mathbb{F}^{|\mathbf{z}_{k,l}|}$ such that $\widehat{Q}_l(\mathbf{z}_{k,l}) = Q_l\left(B_l\mathbf{z}_{k,l} + \mathbf{d}_l\right)$.

Because $\mathrm{var}(Q_l) = \mathbf{z}_{k,l}$, $\widehat{Q}_l(\mathbf{z}_{k,l}) = Q_l\left([B'']_{\mathbf{z}_{k,l}}\mathbf{x} + [\mathbf{d}'']_{\mathbf{z}_{k,l}}\right)$, where $[B'']_{\mathbf{z}_{k,l}}$ and $[\mathbf{d}'']_{\mathbf{z}_{k,l}}$ are B'' and $\mathbf{d}''$ restricted to the rows corresponding to $\mathbf{z}_{k,l}$. Also, since $\mathrm{var}\left(\widehat{Q}_l\right) = \mathbf{z}_{k,l}$, $\widehat{Q}_l(\mathbf{z}_{k,l}) = Q_l\left([B'']_{\mathbf{z}_{k,l} \times \mathbf{z}_{k,l}}\mathbf{z}_{k,l} + [\mathbf{d}'']_{\mathbf{z}_{k,l}}\right)$, where $[B'']_{\mathbf{z}_{k,l} \times \mathbf{z}_{k,l}}$ is B'' restricted to the rows and columns corresponding to $\mathbf{z}_{k,l}$. It follows from Observation 2.2 that $[B'']_{\mathbf{z}_{k,l} \times \mathbf{z}_{k,l}}$ is invertible. So we can set $B_l = [B'']_{\mathbf{z}_{k,l} \times \mathbf{z}_{k,l}}$ and $\mathbf{d}_l = [\mathbf{d}'']_{\mathbf{z}_{k,l}}$.

Thus, by the induction hypothesis, $A_{k,l}$ computed in Step 22, is such that there exist a permutation matrix $P_{k,l} \in M(|\mathbf{z}_{k,l}|, \mathbb{F})$, a scaling matrix $S_{k,l} \in M(|\mathbf{z}_{k,l}|, \mathbb{F})$ and a $\mathbf{b}_{k,l} \in \mathbb{F}^{|\mathbf{z}_{k,l}|}$ satisfying $\widehat{Q}_l(A_{k,l}\mathbf{z}_{k,l}) = Q_l(P_{k,l}S_{k,l}\mathbf{z}_{k,l} + \mathbf{b}_{k,l})$. Since this is true for all $l \in [m_k]$, after the execution of the for loop of lines 20-23 and Step 24, for all $l \in [m_k]$, $c_l\widehat{Q}_{k,l}(A(A_k\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k)) = c_l Q_{k,l}(P_k S_k \mathbf{z}_k + \mathbf{b}_k)$, where for all $l \in [m_k]$ and $z \in \mathbf{z}_{k,l}$, P_k maps z to $P_{k,l} \circ \mathbf{z}_{k,l}$, S_k maps z to $S_{k,l} \circ \mathbf{z}_{k,l}$ and the z-th coordinate of $\mathbf{d}_k$ is the same as that of $[P_{k,0}]_{\mathbf{z}_{k,l} \times \mathbf{z}_{k,l}}\mathbf{d}_{k,l}$. Because $\prod_{l \in [m_k]} c_l = 1$, $\widehat{T}_k(A(A_k\mathbf{z}_k, \mathbf{x} \setminus \mathbf{z}_k)) = T'_k(P_0(P_k S_k \mathbf{z}_k + \mathbf{b}_k, \mathbf{x} \setminus \mathbf{z}_k))$, proving the claim. □

E PE for orbits of product-depth 2 ROFs

In Section 5, we gave an algorithm for PE for orbits of additive-constant-free canonical ROFs. Here we show how to solve PE for product-depth 2 canonical ROFs with additive-constants.

The issue with additive-constants. Let f_1 and f_2 be two n-variate polynomials in the orbits of ROFs and suppose that they are equivalent. Then there exists a canonical ROF $\mathtt{C}$ such that $f_1, f_2 \in \mathrm{orb}(\mathtt{C})$. If $A_1, A_2 \in \mathrm{GL}(n, \mathbb{F})$ are matrices obtained by invoking the Find-Equivalence() algorithm (Algorithm 1) on f_1 and f_2, respectively, then $f_1(A_1\mathbf{x})$, $f_2(A_2\mathbf{x}) \in \text{PS-orb}(\mathtt{C})$. In particular, the additive-constants other than translations in $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$ are the same. However,

when we reconstruct $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$ using the Reconstruct-ROF() algorithm (Algorithm 13) and recover the translation of variables using the Canonize() procedure (Procedure 14), the outputs $\mathtt{C}'_1$ and $\mathtt{C}'_2$ are equal to $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$ up to scaling of the leaves. This means that the additive-constants in $\mathtt{C}'_1$ and $\mathtt{C}'_2$ might not be the same. Thus if we were to construct a permutation matrix P from the isomorphism that maps the underlying tree of $\mathtt{C}'_1$ to that of $\mathtt{C}'_2$, $\mathtt{C}'_1$ need not be equal to $\mathtt{C}'_2(P\mathbf{x})$. So the strategy used in Section 5 does not work in a straightforward manner. In this section, we show how to overcome this issue for the case of orbits of product-depth 2 ROFs.

The idea. Suppose $f_1 \in \text{orb}(f_2)$; then $\mathtt{C}_1 = \mathtt{C}_2 = \mathtt{C}$, where $\mathtt{C}$ is a product-depth 2 canonical ROF and thus, from Theorem 1 $f_1(A_1\mathbf{x}) \in \text{PS-orb}(f_2(A_2\mathbf{x}))$. We reconstruct $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$ to obtain $\mathtt{C}'_1$ and $\mathtt{C}'_2$, recover, and remove the translations of variables in both $\mathtt{C}'_1$ and $\mathtt{C}'_2$. We then show that there is a way to transform $\mathtt{C}'_1$ and $\mathtt{C}'_2$ such that all the non-zero additive-constants in them are 1 and that as ROFs, they only differ by permutation and scaling of variables; we exploit this to give an equivalence test. We then recover the scaling of variables in $\mathtt{C}'_1$ and $\mathtt{C}'_2$. After that we check if for every term T_1 in $\mathtt{C}'_1$ there exists a term T_2 in $\mathtt{C}'_2$ such that the number of factors of both having 1 as additive-constant and having 0 as additive-constant is the same. Furthermore, we check that for every factor of T_1 having 1 (respectively, 0) as additive-constant, there exists a factor of T_2 also having 1 (respectively, 0) as additive-constant such that their underlying trees are isomorphic. If $f_1 \in \text{orb}(f_2)$, this must be true for all terms of $\mathtt{C}'_1$ and $\mathtt{C}'_2$ and we are thus able to check for equivalence. Note that here we do not need to worry about their additive-constants as they are the same.

Transforming $\mathtt{C}'_1$ and $\mathtt{C}'_2$. Let $f(\mathbf{x}) = f_1(\mathbf{x}), A = A_1, \mathtt{C}'(\mathbf{x}) = \mathtt{C}'_1(\mathbf{x})$ (or $f(\mathbf{x}) = f_2(\mathbf{x}), A = A_2, \mathtt{C}'(\mathbf{x}) = \mathtt{C}'_2(\mathbf{x})$). Suppose that $f(A\mathbf{x}) = T_1 + \cdots + T_s + \gamma$. Then from Lemma F.1 as $\mathtt{C}'$ and $f(A\mathbf{x})$ are equal up to scaling of leaves and each gate in $\mathtt{C}'$ computes a non-zero constant multiple of the corresponding gate in $f(A\mathbf{x})$, if $\mathtt{C}'(\mathbf{x}) = T'_1 + \cdots + T'_s + \gamma'$, then $T'_i = c_i T_i$ for all $i \in [s]$ and $\gamma' = c_0\gamma$, where $c_0, \ldots, c_s$ are non-zero constants.

Observation E.1. $c_0, \ldots, c_s = 1$.

The proof of the above observation can be found in Section E.1. Let $\mathbf{b}$ be the translation vector output by Canonize($\mathtt{C}'$). Then, from Claim F.3, $\mathbf{b}$ is also the translation vector of $f(A\mathbf{x})$. So, $f(A\mathbf{x} + \mathbf{b})$ is free of translations and is the same as $\mathtt{C}'(\mathbf{x} + \mathbf{b})$ up to scaling of the leaves. We shall transform the terms of $\mathtt{C}'(\mathbf{x} + \mathbf{b})$. Let $T = Q_1 \cdots Q_m$ be a term of $f(A\mathbf{x} + \mathbf{b})$. Then, from Lemma F.1 the corresponding term of $\mathtt{C}'$, $T' = Q'_1 \cdots Q'_m$ will be such that $Q'_i = \beta_i \cdot Q_i$, $\beta_i \neq 0$ for all $i \in [m]$. There are two kinds of T':

- Kind 1: The additive-constant of at least one of the factors $Q'_1, \ldots, Q'_m$ is 0.
- Kind 2: The additive-constant of all the factors $Q'_1, \ldots, Q'_m$ are non-zero.

In the following claims we see how to transform each of the above two kinds of terms. Their proofs can be found in Sections E.2 and E.3.

Claim E.1. *Let T' be a term of kind 1 such that for some $k < m$, the additive-constants of $Q'_{k+1}, \ldots, Q'_m$ are zero, while the additive-constants $\alpha'_1, \ldots, \alpha'_k$ of $Q'_1, \ldots, Q'_k$ are non-zero. Also, let $\alpha_1, \ldots, \alpha_m$ be the additive-constants in T. Then, if we transform T' by bringing out $\alpha'_1, \ldots, \alpha'_k$ and absorb the product $\alpha' = \prod_{i\in[k]} \alpha'_i$ in Q'_m, we recover T' as $\frac{Q_1}{\alpha_1} \cdots \frac{Q_k}{\alpha_k} \cdot (\beta_{k+1}Q_{k+1}) \cdots (\beta_{m-1}Q_{m-1}) \cdot (\beta \cdot \alpha \cdot \beta_m Q_m)$, where $\beta = \prod_{i\in[k]} \beta_i$ and $\alpha = \prod_{i\in[k]} \alpha_i$. Also, the only non-zero additive-constants in T' are all 1.*

Claim E.2. *Let T' be a term of kind 2 and the additive-constants of $Q'_1, \ldots, Q'_m$ be $\alpha'_1, \ldots, \alpha'_m$. Also, let $\alpha_1, \ldots, \alpha_m$ be the additive-constants of T. Then, if we transform T' by bringing out $\alpha'_1, \ldots, \alpha'_m$, we recover T' as $\alpha \cdot \frac{Q_1}{\alpha_1} \cdots \frac{Q_m}{\alpha_m}$, where $\alpha = \prod_{i \in [m]} \alpha_i$. Also, all additive-constants in T' are 1.*

Now, suppose that $f_1 \in \text{orb}(f_2)$ and $\mathbf{b}_1, \mathbf{b}_2$ are translation vectors of $\mathtt{C}'_1$ and $\mathtt{C}'_2$ recovered using the Canonise procedure. Then from Claim F.3, as they are also translation vectors of $f_1(A_1\mathbf{x})$ and $f_2(A_2\mathbf{x})$, $f_1(A_1\mathbf{x} + \mathbf{b}_1)$ and $f_2(A_2\mathbf{x} + \mathbf{b}_2)$ are free of translations. Moreover, as ROFs they only differ by permutation and scaling of variables. Notice that, in this case, the above two claims imply that the same relationship also holds between $\mathtt{C}'_1(\mathbf{x} + \mathbf{b}_1)$ and $\mathtt{C}'_2(\mathbf{x} + \mathbf{b}_2)$. As we exploit this property to give an equivalence test for f_1 and f_2, we record it as an observation.

Observation E.2. *If $f_1 \in \text{orb}(f_2)$, then after modifying $\mathtt{C}'_1(\mathbf{x} + \mathbf{b}_1)$ and $\mathtt{C}'_2(\mathbf{x} + \mathbf{b}_2)$ according to Claims E.1 and E.2, $\mathtt{C}'_1(\mathbf{x} + \mathbf{b}_1)$ and $\mathtt{C}'_2(\mathbf{x} + \mathbf{b}_2)$ as ROFs only differ by permutation and scaling of variables.*

The Algorithm

Algorithm 10 Product-Depth-2-Equivalence-Test($f_1(\mathbf{x}), f_2(\mathbf{x})$)

Input: Black-box access to $f_1(\mathbf{x}), f_2(\mathbf{x})$ in the orbits of product-depth 2 canonical ROFs.
Output: Whether or not f_1 and f_2 are equivalent. If they are equivalent, then $A \in \text{GL}(n, \mathbb{F})$ and $\mathbf{b} \in \mathbb{F}^n$ such that $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$.

/* Reconstructing canonical ROFs equivalent to f_1 and f_2 and transforming their terms. */

1. **for** $i \in [2]$ **do**
2. $A_i \leftarrow$ Find-Equivalence($f_i(\mathbf{x})$) (Algorithm 1).
3. $\mathtt{C}'_i \leftarrow$ Reconstruct-ROF($f_i(A_i\mathbf{x})$) (Algorithm 13).
4. $\mathbf{b}_i \leftarrow$ translation vector returned by Canonize($\mathtt{C}'_i$) (Procedure 14). $\mathtt{C}'_i \leftarrow \mathtt{C}'_i(\mathbf{x} + \mathbf{b}_i)$.
5. Transform all terms in $\mathtt{C}'_i$ according to Claims E.1 and E.2. $\mathtt{C}'_i \leftarrow$ the ROF obtained after the transformation and recovering scaling of variables, $S_i \leftarrow$ the scaling matrix.
6. **end for**

/* Checking if $\mathtt{C}'_1$ and $\mathtt{C}'_2$ are equivalent. */

7. **if** the additive-constants of $\mathtt{C}'_1$ and $\mathtt{C}'_2$ attached to the top $+$ gate are not equal **then**
8. Return NOT EQUIVALENT.
9. **end if**
10. $N_1 \leftarrow$ set of terms of $\mathtt{C}'_1$, $N_2 \leftarrow$ set of terms of $\mathtt{C}'_2$, $P \leftarrow I_{n \times n}$, the permutation matrix mapping the variables of $\mathtt{C}'_1$ to the variables of $\mathtt{C}'_2$.
11. **for** $T'_1 \in N_1$ **do**
12. If T'_1 is a term of kind 1 and $\exists\ T'_2 \in N_2$ also of kind 1 such that Check-Kind-1(T'_1, T'_2) returns SUCCESS, then $N_2 \leftarrow N_2 \setminus \{T'_2\}$. Update P so that it maps var(T'_1) to var(T'_2) appropriately.
13. If T'_1 is a term of kind 2 and $\exists\ T'_2 \in N_2$ also of kind 2 such that Check-Kind-2(T'_1, T'_2) returns SUCCESS, then $N_2 \leftarrow N_2 \setminus \{T'_2\}$. Update P so that it maps var(T'_1) to var(T'_2) appropriately.
14. **end for**
15. **if** $N_2 = \emptyset$ **then**
16. $A \leftarrow A_2 S_2 P S_1^{-1} A_1^{-1}$, $\mathbf{b} \leftarrow A_2\mathbf{b}_2 - A_2 S_2 P S_1^{-1}\mathbf{b}_1$.
17. Use the Schwartz-Zippel Lemma to check if $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$. If yes, return EQUIVALENT, A and $\mathbf{b}$. Else, return NOT EQUIVALENT.

18. **else**
19. Return NOT EQUIVALENT.
20. **end if**

The checks on lines 12 and 13 are performed using the following procedures.

Procedure 11 Check-Kind-1(T_1', T_2')

Input: Terms T_1' of $\mathtt{C}_1'$ and T_2' of $\mathtt{C}_2'$ of Kind 1.
Output: SUCCESS if they are equivalent, FAILURE otherwise.

1. Suppose $T_1' = Q_{1,1}' \cdots Q_{1,k_1}' \cdot Q_{1,k_1+1}' \cdots Q_{1,m_1}'$ and $T_2' = Q_{2,1}' \cdots Q_{2,k_2}' \cdot Q_{2,k_2+1}' \cdots Q_{2,m_2}'$, where $Q_{1,1}', \cdots, Q_{1,k_1}'$ and $Q_{2,1}', \cdots, Q_{2,k_2}'$ are the only factors with additive-constants.
2. **if** $k_1 \neq k_2$ or $m_1 \neq m_2$ **then**
3. Return FAILURE
4. **end if**
5. **if** there exists a bijection $\sigma : [m_1] \to [m_1]$ such that $\sigma([k_1]) = [k_1]$ and $\forall i \in [m_1]$, the rooted trees of $Q_{1,i}'$ and $Q_{2,\sigma(i)}'$ are isomorphic **then**
6. Return SUCCESS.
7. **else**
8. Return FAILURE.
9. **end if**

Procedure 12 Check-Kind-2(T_1', T_2')

Input: Terms T_1' of $\mathtt{C}_1'$ and T_2' of $\mathtt{C}_2'$ of kind 2.
Output: SUCCESS if they are equivalent, FAILURE otherwise.

1. Suppose $T_1' = \alpha_1' \cdot Q_{1,1}' \cdots Q_{1,m_1}'$ and $T_2' = \alpha_2' \cdot Q_{2,1}' \cdots Q_{2,m_2}'$.
2. **if** $\alpha_1' \neq \alpha_2'$ or $m_1 \neq m_2$ **then**
3. Return FAILURE
4. **end if**
5. **if** there exists a bijection $\sigma : [m_1] \to [m_1]$ such that $\forall i \in [m_1]$, the rooted trees of $Q_{1,i}'$ and $Q_{2,\sigma(i)}'$ are isomorphic **then**
6. Return SUCCESS.
7. **else**
8. Return FAILURE.
9. **end if**

Analysis of the algorithm

We establish the correctness of the above algorithm by proving the following lemma.

Lemma E.1 (Correctness of Algorithm 10)**.** *Given black-box access to two n-variate polynomials $f_1(\mathbf{x})$, $f_2(\mathbf{x})$ in the orbits of two unknown product-depth 2 canonical ROFs, Algorithm 10 correctly determines whether they are equivalent or not provided that* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\geq n^2$ *and* $|\mathbb{F}| \geq n^{13}$*. Moreover, if they are equivalent, it returns an $A \in \mathrm{GL}(n, \mathbb{F})$ and a $\mathbf{b} \in \mathbb{F}^n$ such that $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$.*

Proof. If $f_1 \notin \text{orb}(f_2)$, then Step 17 ensures that the algorithm returns NOT EQUIVALENT with high probability. So suppose that $f_1 \in \text{orb}(f_2)$. Then, from Observation E.2, we have that after Step 5 of the algorithm, $\mathtt{C}'_1$ and $\mathtt{C}'_2$ as ROFs only differ by permutation of variables (because the scaling of variables has already been recovered). Thus, if the additive-constants attached to the top-most gates in $\mathtt{C}'_1$ and $\mathtt{C}'_2$ are not equal, $f_1 \notin \text{orb}(f_2)$ and so Step 8 is correct.

Now for every term T_1 of f_1, there must exist a term T_2 of f_2 such that $T_1 \in \text{PS-orb}(T_2)$. Then, Observation E.2 also implies that the corresponding terms T'_1 and T'_2 of f'_1 and f'_2 as ROFs must be same up to permutation of variables. It is easy to see that this is true if and only if, depending on the kind of these terms, either Check-Kind-1(T'_1, T'_2) or Check-Kind-2(T'_1, T'_2) succeeds. Hence the algorithm correctly determines whether f_1 and f_2 are equivalent or not. A simple calculation then shows that $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$ for A and $\mathbf{b}$ as defined in Step 17. □

Running time of the algorithm. Find-Equivalence(), Reconstruct-ROF(), and Canonize() run in time polynomial in n. Also as mentioned in Fact A.4, a polynomial time algorithm exists for the rooted tree isomorphism problem. This implies that Check-Kind-1() and Check-Kind-2() run in time poly(n). As $|N_1|, |N_2| \leq n$, this means that the for loop of lines 11-14 also runs in poly(n) time. Moreover, the Schwartz-Zippel lemma also yields a polynomial time algorithm for checking if $f_1(\mathbf{x}) = f_2(A\mathbf{x} + \mathbf{b})$ in Step 17. Thus, Algorithm 10 runs in time poly(n).

E.1 Proof of Observation E.1

Since $f(A\mathbf{x})$ and $\mathtt{C}'(\mathbf{x})$ are the same polynomials, we get $0 = \mathtt{C}'(\mathbf{x}) - f(A\mathbf{x}) = (c_1 - 1)T_1 + \cdots + (c_s - 1)T_s + (c_0 - 1)\gamma$. Now each of the terms $T_1, \ldots, T_s$ contains a variable not contained in any other term or in γ. This means $T_1, \ldots, T_s, \gamma$ are linearly independent forcing $c_0 = \cdots = c_s = 1$. □

E.2 Proof of Claim E.1

As $Q'_i = \beta_i Q_i$, $\alpha'_i = \beta_i \alpha_i$ for all $i \in [m]$. Thus, when we bring out $\alpha'_1, \ldots, \alpha'_k$ from $Q'_1, \ldots, Q'_k$ we recover T' as $T' = (\beta_1\alpha_1) \cdots (\beta_k\alpha_k) \cdot \frac{\beta_1 Q_1}{\beta_1 \alpha_1} \cdots \frac{\beta_k Q_k}{\beta_k \alpha_k} \cdot (\beta_{k+1}Q_{k+1}) \cdots (\beta_m Q_m)$, which implies $T' = (\beta_1\alpha_1) \cdots (\beta_k\alpha_k) \cdot \frac{Q_1}{\alpha_1} \cdots \frac{Q_k}{\alpha_k} \cdot (\beta_{k+1}Q_{k+1}) \cdots (\beta_m Q_m)$. So, after absorbing $\alpha'_1 \cdots \alpha'_k = (\beta_1\alpha_1) \cdots (\beta_k\alpha_k)$ in Q_m, we get T' in the desired form. Also, the only non-zero additive-constants are those in $\frac{Q_1}{\alpha_1}, \ldots, \frac{Q_k}{\alpha_k}$ and they are 1 by the definition of $\alpha_1, \ldots, \alpha_k$. □

E.3 Proof of Claim E.2

As $Q'_i = \beta_i Q_i$, $\alpha'_i = \beta_i \alpha_i$ for all $i \in [m]$. Thus, when we bring out $\alpha'_1, \ldots, \alpha'_m$ from $Q'_1, \ldots, Q'_m$ we recover T' as $T' = (\beta_1\alpha_1) \cdots (\beta_m\alpha_m) \cdot \frac{\beta_1 Q_1}{\beta_1 \alpha_1} \cdots \frac{\beta_m Q_m}{\beta_m \alpha_m}$. Observation E.1 implies $\beta_1 \cdots \beta_m = 1$ and we get $T' = \alpha \cdot \frac{Q_1}{\alpha_1} \cdots \frac{Q_m}{\alpha_m}$. By the definition of $\alpha_1, \ldots, \alpha_m$ all additive-constants in T' are 1. □

F ROF reconstruction

We present an algorithm that reconstructs an ROF in the PS-orb of a canonical ROF.[38] In this section, we slightly abuse the terminology and call an ROF that satisfies Properties 1-5 of Definition

[38] The algorithm can be easily adapted to work for a general ROF. However, since we only need to reconstruct ROFs in the PS-orb of a canonical ROF, we present the algorithm and its analysis just for ROFs in this form.

2.6, but does not necessarily satisfy Property 6, a canonical ROF. We also give an accompanying procedure to recover a translation vector and a scaling matrix that convert the reconstructed ROF to a canonical ROF. While randomized [HH91, BHH95a] and deterministic [SV14, MV18] polynomial-time ROF reconstruction algorithms are known, we provide a randomized algorithm here as we need some special properties of this algorithm in Section 5 and Appendix E.

F.1 The algorithm

Before formally describing the algorithm, let us see a high-level description of it.

The idea. Suppose that we have black-box access to an ROF $\mathtt{C} = T_1 + \cdots + T_s + \gamma$ in the PS-orb of a canonical ROF, where $T_k = Q_{k,1} \cdots Q_{k,s_k}$ for all $k \in [s]$, $Q_{k,l}$ is either a variable or a +-rooted sub-ROF of $\mathtt{C}$ for every $l \in [s_k]$, and $\gamma \in \mathbb{F}$. We first use the second-order derivatives of $\mathtt{C}$ to learn $\mathrm{var}(T_1), \ldots, \mathrm{var}(T_s)$. Then, we obtain black-box access to $T_1, \ldots, T_s$ as follows: As $\mathtt{C}$ is in the PS-orb of a canonical ROF, at most one of the T_k, say T_s, is a scalar multiple of a variable x_i. As $\frac{\partial \mathtt{C}}{\partial x_i x_j} = 0$ for all $j \neq i \in [n]$, we can find out x_i. On the other hand, for $k \in [s-1]$ and for a $x_i \in \mathrm{var}(Q_{k,l})$,

$$\frac{\partial \mathtt{C}}{\partial x_i} = r_1 \cdots r_m \cdot \prod_{l' \in [s_k] \setminus \{l\}} Q_{k,l'},$$

where $r_1, \cdots, r_m$ are pairwise variable disjoint, and every r_i is either a variable or a $+$-rooted sub-ROF of $Q_{k,l}$. As $s_k \geq 2$ and $Q_{k,1}, \ldots, Q_{k,s_k}$ are non-constant polynomials, for every $Q_{k,l}$, there exists $x_j \in \mathrm{var}(T_k)$ such that $Q_{k,l}$ is an irreducible factor of $\frac{\partial \mathtt{C}}{\partial x_j}$ (because of Fact 2.5). Thus, by obtaining black-box access to the derivatives of $\mathtt{C}$ with respect to the variables in $\mathrm{var}(T_k)$, factoring them, collecting all the factors and then discarding a factor r if there exists another factor r' such that $\mathrm{var}(r) \subset \mathrm{var}(r')$, we get black-box access to $Q_{k,1}, \ldots, Q_{k,s_k}$ up to constant multiples. However, notice that if we want to query the black-box of a $Q_{k,l}$ at one point, we need to query the black-box of $\mathtt{C}$ at $\mathrm{poly}(n)$ points. So, if we try to recursively learn $Q_{k,1}, \ldots, Q_{k,s_k}$, the running time of the algorithm would be exponential in the depth of $\mathtt{C}$.

We need to be able to get black-box access to $Q_{k,1}, \ldots, Q_{k,s_k}$ in such a way that to obtain the value of $Q_{k,l}$ at one point, we only need to query the black-box of $\mathtt{C}$ at one point. We do this by first learning $\mathrm{var}(Q_{k,1}), \ldots, \mathrm{var}(Q_{k,s_k})$ and some roots of $Q_{k,1}, \ldots, Q_{k,s_k}$ using the black-boxes of $Q_{k,1}, \ldots, Q_{k,s_k}$ that we obtained above. Then we set all variables in $\mathbf{x} \setminus \mathrm{var}(Q_{k,l})$ to random field elements. This gives us black-box access to $c_{k,l} Q_{k,l} + c'_{k,l}$, for some unknown $c_{k,l} \neq 0, c'_{k,l} \in \mathbb{F}$. Plugging in the root of $Q_{k,l}$ into this black-box we learn $c'_{k,l}$. Subtracting this from $c_{k,l} Q_{k,l} + c'_{k,l}$ gives us black-box access to $c_{k,l} Q_{k,l}$, where $c_{k,l}$ is unknown. Notice that now we only need to make one query to the black-box of $\mathtt{C}$ to learn the value of $c_{k,l} Q_{k,l}$.

We learn γ by finding a common root $\mathbf{a} = (a_1, \ldots, a_n)$ of $Q_{k,l}$ for all $k \in [s], l \in [s_k]$ and setting $\gamma = \mathtt{C}(\mathbf{a})$. Then, we find out $c_1 \ldots, c_s \in \mathbb{F}$ such that $\sum_{k=1}^{s} c_k \cdot \prod_{l=1}^{s_k} c_{k,l} \cdot Q_{k,l} = \mathtt{C} - \gamma$ and multiply $Q_{1,1}, \ldots, Q_{s,1}$ by $c_1, \ldots, c_s$, respectively. After that, we learn T_k by recursively learning $c_k c_{k,1} Q_{k,1}, c_{k,2} Q_{k,2}, \ldots, c_{k,s_k} Q_{k,s_k}$ and multiplying them. We now describe the algorithm formally.

Algorithm 13 Reconstruct-ROF($f(\mathbf{x})$)

Input: Black-box access to an n-variate ROF $f(\mathbf{x})$ in the PS-orb of a canonical ROF.
Output: An ROF $\mathtt{C}$ in the PS-orb of a canonical ROF computing f.

/* Learning $\mathrm{var}(T_1), \ldots, \mathrm{var}(T_s)$. */

1. Let $E \leftarrow \emptyset$, and $G \leftarrow (\mathbf{x}, E)$ be an undirected graph.
2. **for** $i, j \in [|\mathbf{x}|]$ **do**
3. If $\frac{\partial^2 f}{\partial x_i \partial x_j} \neq 0$, add edge $\{x_i, x_j\}$ to E.
4. **end for**
5. Let $\mathscr{C} \leftarrow \{\mathbf{x}_1, \ldots, \mathbf{x}_s\}$ be the set of connected components of G, where $s \leftarrow |\mathscr{C}|$.

/* Discovering factors of $T_1, \ldots, T_s$. */

6. **if** $\exists k \in [s]$ such that $|\mathbf{x}_k| = 1$ **then**
7. If $\mathbf{x}_k = \{x_i\}$, $T_k \leftarrow x_i$, $N_k \leftarrow \{x_i\}$.
8. **end if**
9. **for** $k \in [s]$ such that $|\mathbf{x}_k| \geq 2$ **do**
10. $N_k \leftarrow \emptyset$.
11. **for** i such that $x_i \in \mathbf{x}_k$ **do**
12. Compute black-box access to $\frac{\partial f}{\partial x_i}$ and then obtain black-box access to all its irreducible factors. Add all the irreducible factors to N_k.
13. **end for**
14. **for** $r_1, r_2 \in N_k$ **do**
15. If $\mathrm{var}(r_1) \subseteq \mathrm{var}(r_2)$, $N_k \leftarrow N_k \setminus \{r_1\}$. Else, if $\mathrm{var}(r_2) \subseteq \mathrm{var}(r_1)$, $N_k \leftarrow N_k \setminus \{r_2\}$.
16. **end for**
17. **end for**

/* Obtaining efficient black-box access to the factors of $T_1, \ldots, T_s$. */

18. **for** $k \in [s]$ such that $|\mathbf{x}_k| \geq 1$ **do**
19. **for** $r \in N_k$ **do**
20. $\mathbf{a}_r \leftarrow$ a vector of size $|\mathrm{var}(r)|$ which is a root of r. $\mathbf{a}'_r \leftarrow$ a random vector of size $n - |\mathrm{var}(r)|$.
21. $\beta_r \leftarrow f\left(\mathrm{var}(r) = \mathbf{a}_r,\ \mathbf{x} \setminus \mathrm{var}(r) = \mathbf{a}'_r\right)$. $r \leftarrow f\left(\mathrm{var}(r), \mathbf{x} \setminus \mathrm{var}(r) = \mathbf{a}'_r\right) - \beta_r$.
22. **end for**
23. **end for**

/* Learning γ and $c_1, \ldots, c_s$. */

24. Construct $\mathbf{a} = (a_1, \ldots, a_n)$, a common root of $\prod_{r \in N_1} r, \ldots, \prod_{r \in N_s} r$. $\gamma \leftarrow f(\mathbf{a})$.
25. Solve for $c_1, \ldots, c_s$ such that $f - \gamma = c_1 \cdot \prod_{r \in N_1} r + \cdots + c_s \cdot \prod_{r \in N_s} r$.
26. For all $k \in [s]$, replace an arbitrary $r \in N_k$ by $c_k \cdot r$. If $\exists k \in [s]$ such that $|\mathbf{x}_k| = 1$, $T_k \leftarrow c_k T_k$.

/* Reconstructing $T_1, \ldots, T_s$. */

27. **for** $k \in [s]$ such that $|\mathbf{x}_k| \geq 2$ **do**
28. $T_k \leftarrow 1$.
29. **for** $r \in N_k$ **do**
30. $\mathbf{y} \leftarrow \mathrm{var}(r)$. $T_k \leftarrow T_k \times$Reconstruct-ROF$(r(\mathbf{y}))$.
31. **end for**
32. **end for**
33. $\mathtt{C} \leftarrow T_1 + \cdots + T_s + \gamma$. Return $\mathtt{C}$.

F.2 Analysis of the algorithm

We will assume, without loss of generality, that an ROF has no edge labels and every leaf node is either a constant or a constant multiple of a variable.

Lemma F.1. *If $f(\mathbf{x})$ computed by an ROF $\mathtt{C}'$ in the* PS-orb *of a canonical ROF, then the ROF $\mathtt{C}$ returned by the algorithm is equal to $\mathtt{C}'$ up to scaling of the leaves with high probability. Moreover, there is a one-to-one correspondence between the gates of $\mathtt{C}$ and the gates of $\mathtt{C}'$ with a gate of $\mathtt{C}$ computing a non-zero constant multiple of the polynomial computed by the corresponding gate of $\mathtt{C}'$.*

Proof. We induct on the product-depth Δ of $\mathtt{C}'$. If $\Delta = 0$, then as $\mathtt{C}'$ is in the PS-orb of a canonical ROF, $\mathtt{C}' = c_i x_i + \gamma$, where $c_i \neq 0, \gamma \in \mathbb{F}$. In this case $\mathscr{C}$ has only one connected component $\mathbf{x}_1 = \{x_i\}$. In Step 7, $T_1 = x_i$. Step 24 can be implemented by simply setting $\mathbf{a}$ to be the all zero vector and Step 25 by computing $\frac{\partial f}{\partial x_i}$. Thus, $\mathtt{C} = \mathtt{C}'$ and for $\Delta = 0$, the lemma is true.

Now, suppose that the lemma is true for all ROFs of product-depth at most $\Delta \geq 0$ and let $\mathtt{C}'$ be a product-depth $\Delta + 1$ ROF. Let $\mathtt{C}' = T'_1 + \cdots + T'_s + \gamma'$. There exists at most one $k' \in [s]$, such that $|\mathrm{var}(T'_{k'})| = 1$ and for every $k \in [s] \setminus \{k'\}$, let $T'_k = Q'_{k,1} \cdots Q'_{k,s_k}$, where $s_k \geq 2$ and for every $l \in [s_k]$, $Q'_{k,l}$ is either a variable or a $+$-rooted sub-ROF of $\mathtt{C}'$.

Claim F.1. *There is a bijection $\pi : [s] \to [s]$ s.t. the connected component $\mathbf{x}_{\pi(k)} = \mathrm{var}(T'_k)$ for all $k \in [s]$.*

Proof. Fix any $k \in [s]$. If $|\mathrm{var}(T'_k)| = 1$, say $T'_k = c \cdot x_i$ for some $c \in \mathbb{F}^\times$, then $\frac{\partial^2 f}{\partial x_i \partial x_j} = 0$ for all $j \in [s] \setminus \{i\}$, and so, $\{x_i\}$ is a connected component in $\mathscr{C}$. If $|\mathrm{var}(T'_k)| \geq 2$, then as $\mathtt{C}'$ is in the PS-orb of a canonical ROF, $T'_k = Q'_{k,1} \cdots Q'_{k,s_k}$, where $s_k \geq 2$. If $x_i, x_j \in \mathrm{var}(T'_k)$ are such that $x_i \in \mathrm{var}(Q'_{k,l})$ and $x_j \in \mathrm{var}(Q'_{k,l'})$, for $l \neq l'$, then as $\frac{\partial^2 f}{\partial x_i \partial x_j} \neq 0$, $\{x_i, x_j\} \in E$. On the other hand, if $l = l'$, then as $s_k \geq 2$ and $Q'_{k,1}, \ldots, Q'_{k,s_k}$ are non-constant, there exists a $x_m \in \mathrm{var}(Q'_{k,l''})$, $l'' \neq l$ such that $\frac{\partial^2 f}{\partial x_i \partial x_m} \neq 0$ and $\frac{\partial^2 f}{\partial x_j \partial x_m} \neq 0$. Thus, $\{x_i, x_m\}, \{x_j, x_m\} \in E$ and so x_i and x_j are in the same connected component. Moreover, as for any $x_i \in \mathrm{var}(T'_k)$ and $x_j \in \mathrm{var}(T'_1) \uplus \cdots \uplus \mathrm{var}(T'_{k-1}) \uplus \mathrm{var}(T'_{k+1}) \uplus \cdots \uplus \mathrm{var}(T'_s)$, $\frac{\partial^2 f}{\partial x_i \partial x_j} = 0$, this connected component is exactly $\mathrm{var}(T'_k)$, proving the claim. □

Claim F.2. *After Steps 6-8 and the for loop of lines 18-23 have been executed, for all $k \in [s]$, with high probability, $N_{\pi(k)} = \{Q_{k,1}, ..., Q_{k,s_k}\}$ where $Q_{k,l}$ is a non-zero constant multiple of $Q'_{k,l}$ for all $l \in [s_k]$ and π is the bijection given in Claim F.1.*

Proof. Fix any $k \in [s]$. If $|\mathrm{var}(T'_k)| = 1$, say $\mathrm{var}(T'_k) = \{x_i\}$, then Claim F.1 immediately implies that after Steps 6-8 have been executed, $N_{\pi(k)} = \{x_i\}$. On the other hand if $|\mathrm{var}(T'_k)| \geq 2$, then observe that for any $x_i \in \mathrm{var}(Q'_{k,l})$,

$$\frac{\partial \mathtt{C}'}{\partial x_i} = r_1 \cdots r_m \cdot \prod_{l' \in [s_k] \setminus \{l\}} Q'_{k,l'},$$

where $r_1, \cdots, r_m$ are pairwise variable disjoint and every r_i is a variable or a $+$-rooted sub-ROF of $Q'_{k,l}$. Hence, after the for loop 11-13 has been executed, $N_{\pi(k)}$ will contain two types of factors:

- constant multiples of $Q'_{k,1}, \ldots, Q'_{k,s_k}$ and
- constant multiples of $+$-rooted sub-ROFs of $Q'_{k,1}, ..., Q'_{k,s_k}$.

The first kind of factors are present because $s_k \geq 2$, $Q'_{k,1}, ..., Q'_{k,s_k}$ are non-constant polynomials and being $+$-rooted sub-ROFs are irreducible (see Fact 2.5). As all variables appearing in any $+$-rooted

sub-ROF of $Q'_{k,l}$ are also variables of $Q'_{k,l}$, the second kind of factors are removed from $N_{\pi(k)}$ by the for loop of lines 14-16. Moreover, as $Q'_{k,l}$ and $Q'_{k,l'}$ are variable disjoint for $l \neq l'$, the first kind of factors are not removed. This means that after the for loop of lines 9-17 has been executed, for all $k \in [s]$, $N_{\pi(k)} = \{Q_{k,1}, ..., Q_{k,s_k}\}$ where $Q_{k,l}$ is a non-zero constant multiple of $Q'_{k,l}$ for all $l \in [s_k]$. Thus, a root of $Q_{k,l}$ is also a root of $Q'_{k,l}$.

Now, inside the loop of lines 18-23, for $r = Q_{k,l}$, $f(\text{var}(r), \mathbf{x} \setminus \text{var}(r) = \mathbf{a}'_r) = c_{k,l}Q'_{k,l} + c'_{k,l}$ for some $c_{k,l}, c'_{k,l} \in \mathbb{F}$. As every coordinate of $\mathbf{a}'_r$ is chosen randomly (say, from a subset of $\mathbb{F}$ of size n^4), with high probability $c_{k,l} \neq 0$. As $\mathbf{a}_r$ is a root of $Q'_{k,l}$, $\beta_r = f(\text{var}(r) = \mathbf{a}_r, \mathbf{x} \setminus \text{var}(r) = \mathbf{a}'_r) = c'_{k,l}$. Hence, after this loop has been executed $r = c_{k,l}Q'_{k,l}$, proving the claim. □

Thus, for all $k \in [s]$, T'_k is a non-zero constant multiple of the product of the polynomials in $N_{\pi(k)}$. So, for some $c_1, \ldots, c_s \in \mathbb{F}^\times$, $\mathtt{C}' = T'_1 + \cdots T'_s + \gamma' = \sum_{k\in[s]} c_k \cdot \prod_{l\in[s_k]} Q_{k,l} + \gamma$. Since in Step 24, $\mathbf{a}$ is a common root of $Q_{k,l}$, for all $k \in [s]$ and $l \in [s_k]$, $\gamma' = f(\mathbf{a}) = \gamma$. As the polynomials in $\left\{\prod_{l\in[s_k]} Q_{k,l} : k \in [s]\right\}$ are linearly independent, $c_1, \ldots, c_s$ are unique. Once $c_1, \ldots, c_s$ have been learnt in Step 25 and $N_1, \ldots, N_s$ updated in Step 26, we have for all $k \in [s]$, T'_k is equal to the product of the polynomials in $N_{\pi(k)}$. For all $k \in [s]$ and $l \in [s_k]$, as $Q_{k,l}$ is a product-depth Δ ROF in the PS-orb of a canonical ROF, from the induction hypothesis, the output of Reconstruct-ROF$(Q_{k,l})$ is a ROF in the PS-orb of a canonical ROF and is equal to $Q_{k,l}$ up to scaling of the leaves. After the loop 27-32 has been executed, for all $k \in [s]$, $T_{\pi(k)}$ is an ROF in the PS-orb of a canonical ROF and is equal to T'_k up to scaling of the leaves. Hence, $\mathtt{C}' = T'_1 + \cdots + T'_s + \gamma' = T_1 + \cdots + T_s + \gamma = \mathtt{C}$.

For the "moreover" part of the lemma, notice that from the induction hypothesis, we have the desired one-to-one correspondence between gates of the ROF output by Reconstruct-ROF$(Q_{k,l})$ and the gates of $Q_{k,l}$ for all $k \in [s]$ and $l \in [s_k]$. Then, as $T_{\pi(k)} = T'_k$, this yields the desired one-to-one correspondence between the gates of $\mathtt{C}'$ and $\mathtt{C}$. □

Running time of the algorithm

We will show that the algorithm runs in time poly(n). From black-box access to f, a black-box access to $\frac{\partial^2 f}{\partial x_i \partial x_j}$, for any $i, j \in [n]$, can be computed in poly(n) time (Fact A.1). Whether $\frac{\partial^2 f}{\partial x_i \partial x_j}$ is zero or not can be determined in poly(n) time using the Schwartz-Zippel test. Hence G can be constructed in time poly(n). The connected components of G can also be computed in poly(n) time. Clearly, lines 6-8 run in poly(n) time. Now we analyse the runtime of the loop of lines 9-17.

A black-box access to $\frac{\partial f}{\partial x_i}$ can be obtained in poly(n) time from black-box access to f. Once we have black-box access to $\frac{\partial f}{\partial x_i}$, black-box access to its irreducible factors can be computed in poly(n) time using the algorithm in [KT90]. Hence, the for loop of lines 11-13 executes in poly(n) time. For Step 15, var(r_1) and var(r_2) can be determined by obtaining black-box access to the derivatives of r_1 and r_2 with respect to all $\mathbf{x}$ variables and checking using the Schwartz-Zippel lemma which of them are non-zero. Thus this step, and hence, the for loop of lines 14-16 executes in poly(n) time.

Once we have black-box access to all polynomials in $N_1, \ldots, N_s$, for all $k \in [s]$ and all $r \in N_k$, $\mathbf{a}'_r$ and $\mathbf{a}_r$ can be computed in poly(n) time. To construct $\mathbf{a}_r$, first set all but one variable appearing in r to random values. After doing this, r becomes an affine form whose root can be computed easily. Notice that on line 21, we obtain black-box access to $Q_{k,l}$ using *only one* query to f.

The vector $\mathbf{a}$ can be constructed in poly(n) time by just combining all $\mathbf{a}_r$'s constructed in the loop of lines 18-23. To compute $c_1, \ldots, c_s$ in Step 25, we can simply evaluate f and $\prod_{l\in s_k} Q_{k,l}$ for all

$k \in [s]$ at s many random points $\mathbf{b}_1, \ldots \mathbf{b}_s$ and solve the linear system of equations

$$\left\{ f(\mathbf{b}_i) - \gamma = \sum_{k \in [s]} c_k \cdot \prod_{l \in s_k} Q_{k,l}(\mathbf{b}_i) : k \in [s] \right\}$$

for $c_1, \ldots, c_s$. As $\left\{ \prod_{l \in [s_k]} Q_{k,l} : k \in [s] \right\}$ are linearly independent, with high probability, the coefficient matrix of this system will be invertible.

So far we have shown that for each call to the algorithm, the time spent outside the recursive calls on line 30 is $\text{poly}(n)$. Now, given input f, the total number of recursive calls is at most $\text{poly}(n)$. This is because each leaf of the recursion tree corresponds to a distinct variable in $\mathbf{x}$ and whenever Reconstruct-ROF(r) is called from inside Reconstruct-ROF(r'), $\text{var}(r) \subsetneq \text{var}(r')$. Thus, the runtime of the algorithm is $\text{poly}(n)$, as a black-box query to r amounts to only one query to f.

F.3 Canonization: Recovering scaling and translation

We shall slightly abuse the terminology in this section and say that a leaf of an ROF is a variable if it is a constant multiple of a variable. This is consistent with our assumption in the previous section that any leaf of an ROF is either a constant multiple of a variable or a constant. To recover the scaling matrix and the translation vector, we use the following procedure.

Procedure 14 Canonize($\mathtt{C}$)

Input: An ROF $\mathtt{C}$ in the PS-orb of a canonical ROF.

Output: A scaling matrix $S \in \text{GL}(|\mathbf{x}|, \mathbb{F})$ and a vector $\mathbf{b}$ such that $\mathtt{C}\left(S\mathbf{x} + \mathbf{b}\right)$ is a canonical ROF.

1. $N_1 \leftarrow$ set of all $+$-gates in $\mathtt{C}$ directly connected to a variable. $N_2 \leftarrow$ set of all variable leaves connected to $\times$-gates. Initialize $S \leftarrow I_{n \times n}$ and $\mathbf{b} = (b_1, \ldots, b_n) = \mathbf{0}$.
2. **for** $v \in N_1 \uplus N_2$ **do**
3. If $v \in N_1$ and the variable and constant children of v are $\alpha_i x_i$ and β_i respectively, where $\alpha_i \in \mathbb{F}^\times$, then $b_i \leftarrow \frac{-\beta_i}{\alpha_i}$ and $S \leftarrow \text{diag}\left(0, \ldots, 0, \alpha_i^{-1}, 0, \ldots, 0\right) \cdot S$.
4. Else, if $v \in N_2$ and $v = \alpha_i x_i$, $\alpha_i \in \mathbb{F}^\times$, $S \leftarrow \text{diag}\left(0, \ldots, 0, \alpha_i^{-1}, 0, \ldots, 0\right) \cdot S$ and $b_i \leftarrow 0$.
5. **end for**
6. Return S, $\mathbf{b}$.

Clearly, the procedure runs in $\text{poly}(n)$ time; its correctness follows from the next observation.

Observation F.1. *Let $S, \mathbf{b} = \text{Canonize}(\mathtt{C})$. Then, $\mathtt{C}(S\mathbf{x} + \mathbf{b})$ is a canonical ROF.*

Proof. Let $v \in N_1$ and let the variable and constant children of v be $\alpha_i x_i$ and β_i, respectively. As $b_i = \frac{-\beta_i}{\alpha_i}$ and S is updated as $\text{diag}\left(0, \ldots, 0, \alpha_i^{-1}, 0, \ldots, 0\right) \cdot S$, after the execution of the loop of lines 2-5, $\alpha_i x_i + \beta_i$ from $\mathtt{C}$ becomes x_i in $\mathtt{C}(S\mathbf{x} + \mathbf{b})$. Similarly, if $v \in N_2$ and $v = \alpha_i x_i$, because S is updated as $\text{diag}\left(0, \ldots, 0, \alpha_i^{-1}, 0, \ldots, 0\right) \cdot S$, after the execution of the loop of lines 2-5, $\alpha_i x_i$ becomes x_i in $\mathtt{C}(S\mathbf{x} + \mathbf{b})$. Since the only difference between a canonical ROF and an ROF in its PS-orb is that in the latter a variable can be scaled and translated, this proves the observation. □

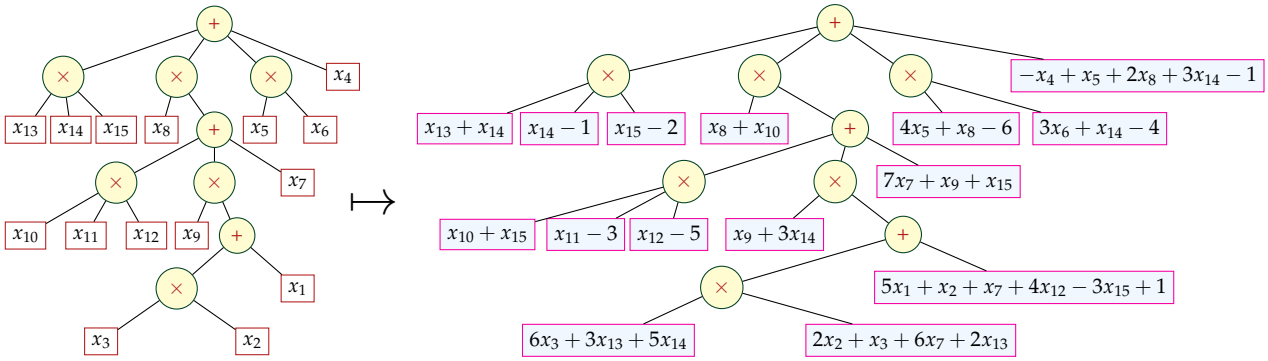
We now show that not only does Procedure 14 recover the translation vector but also that this vector is recovered uniquely. The following claim comes in handy in Appendix E.

Claim F.3. *Let $\mathtt{C}'$ be an ROF in the* PS-orb *of a canonical ROF, S' be a scaling matrix and $\mathbf{b}'$ a translation vector such that $\mathtt{C}'(S'\mathbf{x} + \mathbf{b}')$ is a canonical ROF. Also, let $\mathtt{C}$ = Reconstruct-ROF$(\mathtt{C}'(\mathbf{x}))$ and $S, \mathbf{b}$ = Canonize$(\mathtt{C})$. Then, $\mathbf{b} = \mathbf{b}'$.*

Proof. Let $v \in N_1 \uplus N_2$. From Lemma F.1, the corresponding gate v' in $\mathtt{C}'$ is such that $v = c \cdot v'$ for some $c \neq 0$. So, if $v \in N_1$, then v' must be a $+$ gate with variable and constant children. If the variable and constant children of v' are $\alpha'_i x_i$ and β'_i, respectively, then the variable and the constant children of v are $c\alpha'_i x_i$ and $c\beta'_i$, respectively. Observe that $b'_i = \frac{-\beta'_i}{\alpha'_i}$. Thus, $b_i = \frac{-c\beta'_i}{c\alpha'_i} = \frac{-\beta'_i}{\alpha'_i} = b'_i$. Similarly, if $v \in N_2$, then v' is also a variable leaf. Thus, $b_i = b'_i = 0$. □

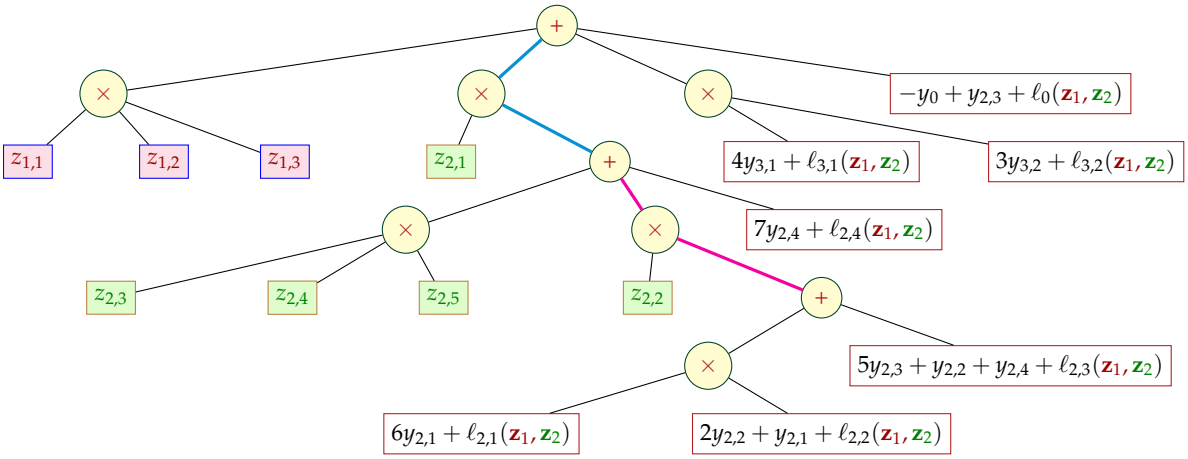
G A pictorial overview of Algorithm 1

In this section, we pictorially depict the execution of Algorithm 1. We consider the following simple example to show the working of Phase 1 of the algorithm mentioned in Section 4.1.



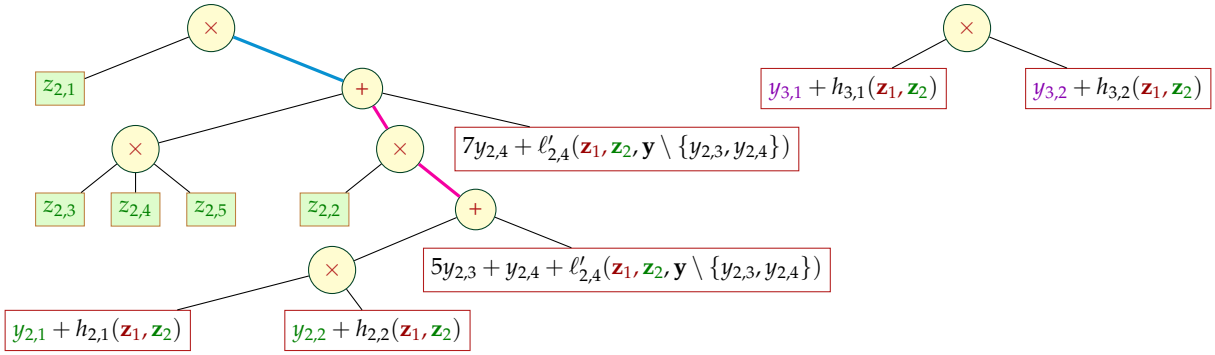
The original ROF $\mathtt{C}$ — $f \in$ orb$(\mathtt{C})$

In the following figures, $\ell_i, \ell_{i,j}, \ell'_{i,j}, h_i, h'_i, h_{i,j}, h'_{i,j}$ etc. denote affine forms.



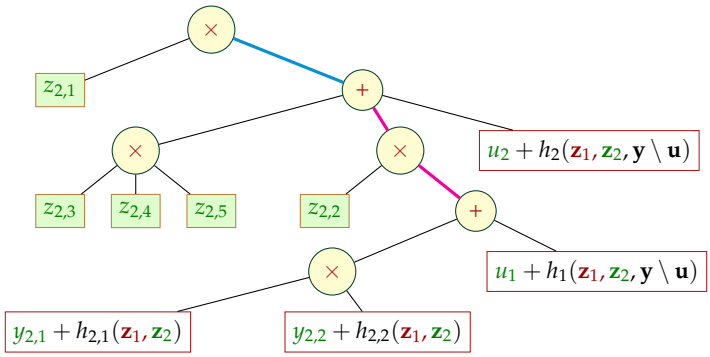
Step 1

In Step 1, all the good terms of f are made variable disjoint. Furthermore, every "good" linear factor - affine form connected to a $\times$ gate computing a polynomial of degree at least 3 in f - gets mapped to (a constant multiple) of a distinct variable. In our example, the good term has three good linear factors, which have been mapped to $z_{1,1}, z_{1,2}, z_{1,3}$. Also, there are five good linear factors in the bad term; these have been mapped to $z_{2,1}, \ldots, z_{2,5}$. Let $\mathbf{z}_1 := \{z_{1,1}, z_{1,2}, z_{1,3}\}$, $\mathbf{z}_2 := \{z_{2,1}, \ldots, z_{2,5}\}$, $\mathbf{z} := \mathbf{z}_1 \uplus \mathbf{z}_2$, and $\mathbf{y} := \mathbf{x} \setminus \mathbf{z}$. Step 2 extensively uses skewed paths. In the above figure, there are two skewed paths identified by the "marker monomials" $z_{2,1}$ and $z_{2,1}z_{2,2}$.



Step 2.1

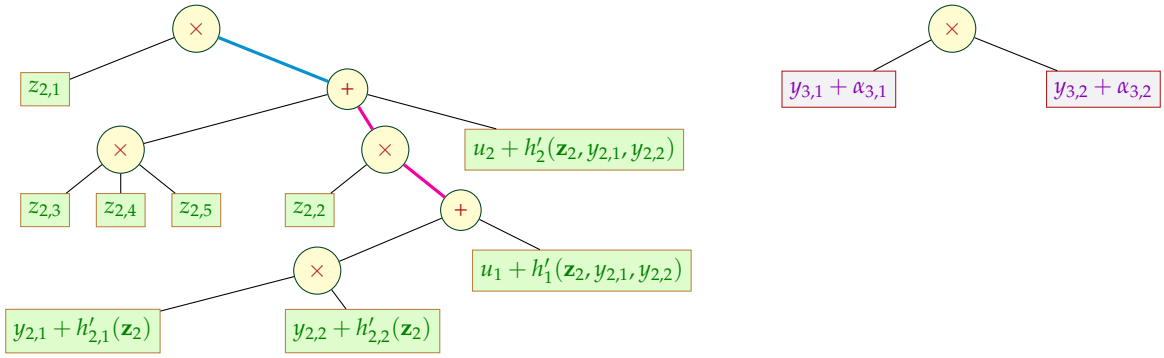
In Step 2.1, every affine form corresponding to a variable in the top quadratic form or a quadratic form along a skewed path which is redundant for $\det(H_C)$ is mapped to an affine form of the type $y_{i,j} + h_{i,j}(\mathbf{z})$. In the above figure, the $\mathbf{y}$-variables corresponding to affine forms in the top quadratic form and in the quadratic form along the skewed path $z_{2,1}z_{2,2}$ are $y_{3,1}, y_{3,2}$ and $y_{2,1}, y_{2,2}$, respectively. We shall refer to all the remaining $\mathbf{y}$-variables, i.e., $y_0, y_{2,3}, y_{2,4}$ as $\mathbf{u}$-variables.



Step 2.2

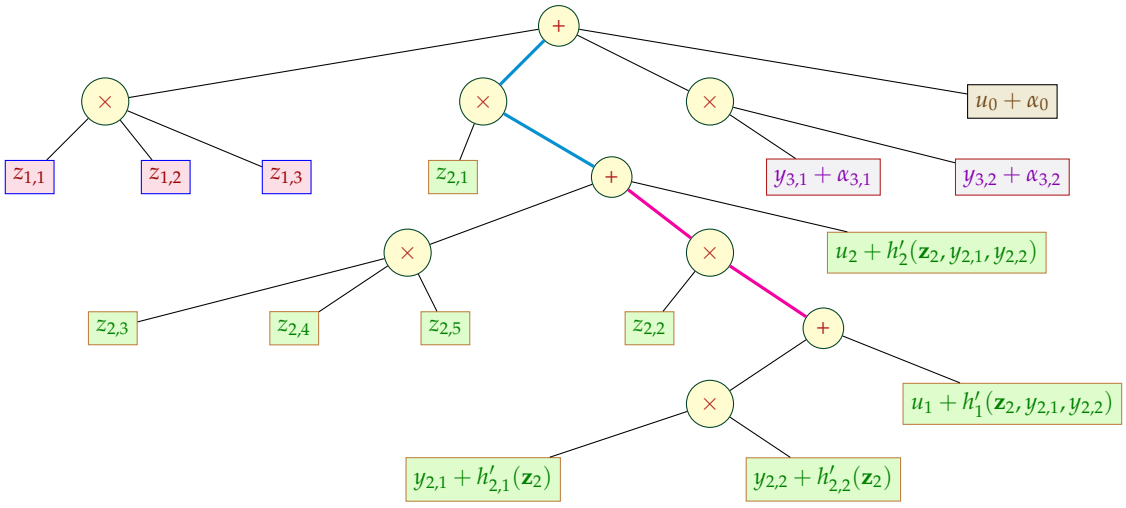
In Step 2.2, every affine form corresponding to a dangling variable along a skewed path which is redundant for $\det(H_C)$ is mapped to an affine form of the type $u_i + h_i(\mathbf{z}, \mathbf{y} \setminus \mathbf{u})$. There might be dangling variables along skewed paths that are present in a set of essential variables of $\det(H_C)$. In our simple example, such variables are not present. In the general case, such variables can be

handled by picking a basis of an appropriate vector space. This space is spanned by the **u**-parts of the affine forms corresponding to the dangling variables along skewed paths and the top dangling variable (see Section 4.1 for more details). A word of caution: the affine form corresponding to the top dangling variable has not been handled; it will be fixed in Step 3. Let $\mathbf{y}_2 = \{u_1, u_2, y_{2,1}, y_{2,2}\}$. In the above figure, the variables in $\mathbf{z}_1$ and $\mathbf{y} \setminus \mathbf{y}_2$ are external for the bad term and all variables in $\mathbf{x} \setminus \{y_{3,1}, y_{3,2}\}$ are external for the top quadratic form. These will be removed in Step 2.3.



Step 2.3

In Step 2.3, external variables are removed from the affine forms in the top quadratic form, quadratic forms along skewed paths and corresponding to dangling variables along skewed paths. In the above figure, $h'_{i,j}$ and h'_i are obtained after removing external variables from $h_{i,j}$ and h_i, respectively.



Step 3

In Step 3, the affine form corresponding to the top-most dangling variable is mapped to $u_0 + \alpha_0, \alpha_0 \in \mathbb{F}$. Now, all the terms of f are variable disjoint. After this, we recursively call Algorithm 1 on the factors of the good and the bad terms to map them to variable disjoint ROFs.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il