On Hardness of Testing Equivalence to Sparse Polynomials Under Shifts

Suryajith Chillara
International Institute of Information Technology, Hyderabad, India

Coral Grichener
Google Research, Herzliya, Israel

Amir Shpilka
Tel Aviv University, Israel

Abstract

We say that two given polynomials $f, g \in R[x_1, \ldots, x_n]$, over a ring R, are equivalent under shifts if there exists a vector $(a_1, \ldots, a_n) \in R^n$ such that $f(x_1 + a_1, \ldots, x_n + a_n) = g(x_1, \ldots, x_n)$. This is a special variant of the polynomial projection problem in Algebraic Complexity Theory.

Grigoriev and Karpinski (FOCS 1990), Lakshman and Saunders (SIAM J. Computing, 1995), and Grigoriev and Lakshman (ISSAC 1995) studied the problem of testing polynomial equivalence of a given polynomial to *any* t-sparse polynomial, over the rational numbers, and gave exponential time algorithms. In this paper, we provide hardness results for this problem.

Formally, for a ring R, let $\mathsf{SparseShift}_R$ be the following decision problem – Given a polynomial $P(X)$, is there a vector $\mathbf{a}$ such that $P(X + \mathbf{a})$ contains fewer monomials than $P(X)$. We show that $\mathsf{SparseShift}_R$ is at least as hard as checking if a given system of polynomial equations over $R[x_1, \ldots, x_n]$ has a solution (Hilbert's Nullstellensatz). As a consequence of this reduction, we get the following results.

1. $\mathsf{SparseShift}_{\mathbb{Z}}$ is undecidable.
2. For any ring R (which is not a field) such that HN_R is NP_R-complete over the Blum-Shub-Smale model of computation, $\mathsf{SparseShift}_R$ is also NP_R-complete. In particular, $\mathsf{SparseShift}_{\mathbb{Z}}$ is also $\mathsf{NP}_{\mathbb{Z}}$-complete.

We also study the gap version of the $\mathsf{SparseShift}_R$ and show the following.

1. For every function $\beta : \mathbb{N} \to \mathbb{R}_+$ such that $\beta \in o(1)$, N^{β}-gap-$\mathsf{SparseShift}_{\mathbb{Z}}$ is also undecidable (where N is the input length).
2. For $R = \mathbb{F}_p, \mathbb{Q}, \mathbb{R}$ or $\mathbb{Z}_q$ and for every $\beta > 1$ the β-gap-$\mathsf{SparseShift}_R$ problem is NP-hard. Furthermore, there exists a constant $\alpha > 1$ such that for every $d = O(1)$ in the sparse representation model, and for every $d \leq n^{O(1)}$ in the arithmetic circuit model, the α^d-gap-$\mathsf{SparseShift}_R$ problem is NP-hard when given polynomials of degree at most d, in $O(nd)$ many variables, as input.

2012 ACM Subject Classification Theory of computation → Circuit complexity; Theory of computation → Algebraic complexity theory

Keywords and phrases algebraic complexity, shift equivalence, polynomial equivalence, Hilbert's Nullstellensatz, hardness of approximation

Digital Object Identifier 10.4230/LIPIcs.STACS.2023.22

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2022/106/`

Funding *Suryajith Chillara*: Part of this work was done while the author was visiting Tel Aviv University, hosted by Amir Shpilka.
Amir Shpilka: The research leading to these results received funding from the Israel Science Foundation (grant number 514/20) and from the Len Blavatnik and the Blavatnik Family foundation.

1 Introduction

This paper studies the following question: given an n-variate polynomial $f(X)$, over a ring R[1], how difficult is the task of finding a shift $\mathbf{b} \in R^n$ such that $f(X+\mathbf{b})$ has fewer monomials than f.

Before proceeding we would like to discuss the issue of representation of polynomials. There are several natural settings – representation as vector of coefficients or as arithmetic circuits – and two different models – the white-box and black-box models. The most obvious representation is the *dense representation* in which n-variate polynomials of degree d are represented as a vectors of coefficients of length $\binom{n+d}{d}$. In this setting we assume that the vector is given as input to the algorithm. A more concise representation is the *sparse representation* in which a polynomial is represented as a list of pairs of exponent vectors and coefficients. In the black-box setting we only assume that the algorithm has black-box access to the polynomial (though the important parameters such as number of variables and degree are known to the algorithm). I.e., the algorithm is restricted to asking the polynomial for its values on different inputs. Another natural model is representing polynomials as arithmetic circuits. That is, the algorithm will get as input an arithmetic circuit computing the polynomial. In the white-box setting the algorithm is explicitly given the circuit so it has access to the graph of computation etc. In the black-box model the algorithm only has black-box access to the circuit (though the important parameters such as size, depth, number of variables etc. are known to the algorithm).

One of the most important questions in the area of Algebraic Complexity Theory is the problem of checking if two polynomials are equivalent under affine transformations. In generality this problem is also called the polynomial projection problem. Ignoring issues of representations the problem is the following.

Polynomial Projection ($\mathsf{PolyProj}_{\mathbb{F}}$):
Given two polynomials $f \in \mathbb{F}[y_1,\ldots,y_m]$ and $g \in \mathbb{F}[x_1,\ldots,x_n]$, over a field $\mathbb{F}$, output an $m \times n$ matrix A and a vector $\mathbf{b} \in \mathbb{F}^m$ such that $g(x_1,\ldots,x_n) = f(A \cdot \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}^T + \mathbf{b})$ if such a pair exists, or output "FAIL" otherwise.

For example, the holy grail of algebraic complexity, Valiant's Extended Hypothesis is an instance of the polynomial projection problem. Recall that the hypothesis says that the permanent of an $n \times n$ matrix cannot be represented as a polynomial projection of determinant of any $m \times m$ matrix, for any m that is polynomial in n [27]. Kayal [19] showed that the problem of polynomial projection is NP-hard in general. However, for specific instances of the polynomial g, under the requirement that the matrix A has full rank (or that it is random), Kayal [19] gave efficient randomized algorithms in the black-box model (i.e. assuming only black-box access to f).

Since studying polynomial equivalence under such projections is NP-hard in general, the following *simpler* question was considered.

Polynomial Equivalence under Shifts ($\mathsf{ShiftEquiv}_{\mathbb{F}}$):
Given two polynomials $f, g \in \mathbb{F}[x_1,\ldots,x_n]$ output a vector $(b_1,\ldots,b_n) \in \mathbb{F}^n$ such that $g(x_1,\ldots,x_n) = f(x_1+b_1,\ldots,x_n+b_n)$ if such a vector exists, or output "FAIL" otherwise.

[1] From now on, R always denotes an integral domain, i.e. a commutative ring with a unit, which is also a domain, and $\mathbb{F}$ a field ($\mathbb{Q}, \mathbb{R}$ and $\mathbb{C}$ are, as usual, the rational, real and complex fields, respectively).

To the best of our knowledge the notion of studying polynomial equivalence under shifts first appeared in [12] and it was formally addressed by Grigoriev in [10]. For polynomials of degree d over n variables, Grigoriev [10] gave a deterministic algorithm over fields of zero characteristic, a randomized algorithm over prime residue fields, and a quantum algorithm over fields of characteristic 2, all of which run in time polynomial in the dense representation. That is, the running time is polynomial in $\binom{n+d}{d}$. If the degree of the polynomial grows as a function of the number of variables or vice versa, the algorithms presented by Grigoriev require *exponential* time in the number of variables, even if the polynomial can be represented by a small arithmetic circuit or if it has polynomially many monomials. It is a natural question to ask if the complexity of the algorithms can be brought down when the input to the algorithm is provided in some succinct representation – for example, as an arithmetic circuit. In such a setting, Dvir, Oliveira and Shpilka [8] showed that given just a black box access to the polynomials f and g on n variables, and given a bound on the degree d and circuit size s, there is a randomized algorithm that runs in time $\text{poly}(n, d, s)$ and solves the polynomial equivalence under shifts problem. The randomness in their algorithm only stems from polynomial identity testing (PIT), which is a sub-routine of their algorithm, and hence equivalence under shifts in this setting can be derandomized if and only if PIT can be derandomized (clearly PIT is a special case of equivalence under shifts when g is the zero polynomial).

A polynomial f is said to be t-sparse if the number of monomials with non-zero coefficients in f is at most t. In the literature, an n variate polynomial is generally said to be sparse if the number of monomials in it with non-zero coefficients is at most $\text{poly}(n)$. Equivalently, a sparse polynomial is a polynomial that can be computed by a depth two $\Sigma\Pi$ arithmetic circuit with a polynomial bound on the top fan-in. Sparse polynomials are extremely well studied because of their simplicity and as a result many efficient algorithmic results are known for them [1, 18, 4, 6, 13, 11, 12, 21, 25].

A variant of the polynomial projection problem asks if a given polynomial is equivalent to a sparse polynomial under affine transformations. This can be seen as a variant of the classical Minimum Circuit Size Problem (MCSP) where given the truth table of a function we wish to find the minimal circuit computing it. In this case the circuit we are seeking is a very structured $\Sigma\Pi\Sigma$ circuit that is obtained by composing a $\Sigma\Pi$ circuit with an affine transformation. As this set of polynomials is dense inside the class $\Sigma\Pi\Sigma$ it is an interesting family to study (see [23]). Grigoriev and Karpinski [12] were the first to consider this variant of the polynomial projection problem. Specifically, they studied the following problem (in the dense representation model) – given a polynomial $P(X)$, over the rationals, and a parameter t output a matrix A and a vector $\mathbf{b}$, if they exist, such that the polynomial $P(A \cdot X + \mathbf{b})$ has at most t monomials. They gave an algorithm whose complexity is $O(M \cdot d^{n^4})$ where M is a bound on the size of coefficients of the input polynomial. Lakshman and Saunders [21] considered the problem of testing the equivalence of univariate polynomials (over $\mathbb{Q}$) to t-sparse polynomials under just shifts instead of affine linear transformations. They provided sufficient conditions for uniqueness and rationality of a t-sparsifying shift. Grigoriev and Lakshman [14] extended these criterion to multivariate polynomials. They also gave algorithms for polynomials with finitely many sparsifying shifts[2] that run in deterministic time $(dt)^{O(n)}$ and randomized time $t^{O(n)}$. In the past two decades, these exponential time algorithms could not be improved and this is a major motivation behind our study of hardness of this problem. We state the following more general problem to allow polynomials over rings.

[2] Over $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ etc., it may happen that there are infinitely many t-sparsifying shifts for a given polynomial. Grigoriev and Lakshman [14] give algorithms for polynomials that are guaranteed to have finitely many t-sparsifying shifts.

Sparsification of Polynomials via Shifts ($\mathsf{SparseShift}_R$):
Given a polynomial $f \in R[x_1, \ldots, x_n]$, decide if there exists a vector $(a_1, \ldots, a_n) \in R^n$ such that $f(x_1 + a_1, \ldots, x_n + a_n)$ has strictly fewer monomials with non-zero coefficients than $f(x_1, \ldots, x_n)$, or output "FAIL" if no such vector exists.

In this paper we show that the problem $\mathsf{SparseShift}_R$ is at least as hard as as checking if a given system of polynomial equations over $R[x_1, \ldots, x_n]$ has a solution (Hilbert's Nullstellensatz).

Hilbert's Nullstellensatz: Given a system S of polynomial equations $S = \{f_1 = 0, \ldots, f_r = 0\}$ over the polynomial ring $R[x_1, \ldots, x_n]$, we say that the system is satisfiable if there exists an assignment $\mathbf{a} \in R^n$ to the variables that simultaneously satisfies all equations in S. This problem has a great significance in Algebraic Geometry and has other important applications in diverse areas. We state a slightly restricted version of Hilbert's Nullstellensatz problem that asks for a common solution in a specific domain (the general version asks for a solution in the algebraic closure). This definition is similar to the definition in the Blum, Shub and Smale model of computation [2].

Hilbert's Nullstellensatz over a ring R (HN_R):
Given a system of polynomial equations $\{f_1 = 0, \ldots, f_r = 0\}$ over $R[x_1, \ldots, x_n]$, decide whether there exist a vector $(a_1, \ldots, a_n) \in R^n$ such that for all $i \in [r]$, $f_i(a_1, \ldots, a_n) = 0$, or output "FAIL" if no such vector exists.

With this background, we shall now state our first main result that gives a reduction from Hilbert's Nullstellensatz problem to polynomial sparsification.

▶ **Theorem 1.** *Let R be an integral domain, which is not a field. Then $\mathsf{SparseShift}_R$ is HN_R-hard, in any of the white-box representations.*

If R is arbitrary, then the polynomials could have coefficients with arbitrary bit complexity. Thus, it is important for us to also specify the model of computation over which this problem is being considered. In the Turing machine model, assuming that $f_1, \ldots, f_r \in \mathbb{C}[x_1, \ldots, x_n]$ have integral coefficients, Koiran [20] showed (by assuming that the Generalized Riemann Hypothesis is true) that $\mathsf{HN}_{\mathbb{C}}$ can be solved in the second level of polynomial hierarchy. Without the GRH assumption, the only known upper bound for $\mathsf{HN}_{\mathbb{C}}$ is PSPACE. For $R = \mathbb{Z}$, Matiyasevich [22] showed that this problem is undecidable (also see [7]). Putting these together with Theorem 1 we get the following consequence.

▶ **Corollary 2.** $\mathsf{SparseShift}_{\mathbb{Z}}$ *is undecidable.*

It is important to note that under sparse or dense representations, $\mathsf{SparseShift}_R$ is in NP_R. That is, given $\mathbf{a}$, we can efficiently verify if it is a sparsifying shift for a polynomial $P(X)$ using at most polynomially many algebraic operations using sparse polynomial interpolation, and sparse polynomial identity testing. Thus for any integral domain R (which is not a field) such that HN_R is NP_R-complete, over the Blum-Shub-Smale model of computation [2], we get the following corollary from the aforementioned statements and Theorem 1.

▶ **Corollary 3.** *Let R be an integral domain (but not a field) such that HN_R is NP_R-complete over the Blum-Shub-Smale model of computation. Then $\mathsf{SparseShift}_R$ is also NP_R-complete.*

In particular, we get that $\mathsf{SparseShift}_{\mathbb{Z}}$ is also $\mathsf{NP}_{\mathbb{Z}}$-complete.

These results, to some extent, shed a light on why this problem in general has been evading the efforts to provide efficient algorithms.

Note that our problem $\mathsf{SparseShift}_R$ can also be viewed as a gap decision problem – given a polynomial $P(X)$ of sparsity t, is there a vector $\mathbf{a}$ such that $P(X+\mathbf{a})$ has at most $t-1$ monomials. Let us formally define a more general gap version of $\mathsf{SparseShift}_R$.

α-gap-$\mathsf{SparseShift}_R$:
Let $\alpha > 1$ be a parameter. Given a polynomial $P \in R[X]$ and a parameter t, ▬ output YES if there exists a vector $\mathbf{a}$ such that $P(X+\mathbf{a})$ has at most t monomials, and ▬ output NO if for all vectors $\mathbf{a}$, $P(X+\mathbf{a})$ has at least αt monomials.

Using gap amplification we reduce $\mathsf{SparseShift}_R$ to N^β-gap-$\mathsf{SparseShift}_R$ for all functions $\beta \in o(1)$. We thus get our second main result.

▶ **Theorem 4.** *For every function $\beta : \mathbb{N} \to \mathbb{R}_+$ such that $\beta \in o(1)$, N^β-gap-$\mathsf{SparseShift}_\mathbb{Z}$ is undecidable (where N is the input length).*

In Theorem 4, we used the undecidability of $\mathsf{HN}_\mathbb{Z}$ to infer the undecidability of N^β-gap-$\mathsf{SparseShift}_\mathbb{Z}$. However, we do not have such results for rings $R \neq \mathbb{Z}$ (over Turing machine model). Furthermore, $\mathsf{HN}_\mathbb{Q}$ is not known to be undecidable and, as mentioned above, over $\mathbb{C}$ it is decidable as well as over finite fields. Thus for $R = \mathbb{F}_p, \mathbb{Q}, \mathbb{R}$ or $\mathbb{Z}_q$, we present a different reduction of gap problems – from $(1-\varepsilon,\delta)$-gap-Max-3Lin$_R$ to α-gap-$\mathsf{SparseShift}_R$ and infer NP-hardness results for α-gap-$\mathsf{SparseShift}_\mathbb{R}$.

$(1-\varepsilon,\delta)$-gap-Max-3Lin$_R$:
Given a system of linear equations $\{L_1 = 0, \ldots, L_m = 0\}$ over $R[x_1, \ldots, x_n]$ each of which depends on exactly 3 variables, ▬ output YES if at least $(1-\varepsilon)$ fraction of equations can be simultaneously satisfied, and ▬ output NO if at most δ fraction of equations can be simultaneously satisfied.

We say that it is NP-hard to $(1-\varepsilon,\delta)$-approximate Max-3Lin$_R$ if the decision problem $(1-\varepsilon,\delta)$-gap-Max-3Lin$_R$ is NP-hard. Using this notion, we summarize non-exhaustively some known NP-hardness results for $(1-\varepsilon,\delta)$-approximating Max-3Lin$_R$.

Table 1 Non-exhaustive list of known NP-hardness results for approximating Max-3Lin$_R$.

Result	Ring R	NP-Hardness for
Håstad [17]	$\mathbb{F}_p$	$\forall\ \varepsilon > 0$, $(1-\varepsilon, \frac{1+\varepsilon}{p})$-approximation
Håstad [17]	$\mathbb{Z}_q$ for $q \in \mathbb{N}$	$\forall\ \varepsilon,\delta > 0$, $(1-\varepsilon, \frac{1}{q}+\delta)$-approximation
Feldman, Gopalan, Khot and Ponnuswami [9]	$\mathbb{Q}$	$\forall\ \varepsilon > 0$, $(1-\varepsilon,\varepsilon)$-approximation
Gurswami and Raghavendra [15, 16]	$\mathbb{Q}, \mathbb{R}$	$\forall\ \varepsilon,\delta > 0$, $(1-\varepsilon,\delta)$-approximation

Thus, a gap reduction from $(1-\varepsilon',\delta')$-gap-Max-3Lin$_R$ (where ε' and δ' are as in the last column of Table 1) to α-gap-$\mathsf{SparseShift}_R$ (for $\alpha = \alpha(\varepsilon',\delta',R)$) implies hardness of α-gap-$\mathsf{SparseShift}_R$ and using amplification we get our third main result. We first state it in the sparse representation model and then in the arithmetic circuit model.

▶ **Theorem 5** (Sparse representation). *For $R = \mathbb{F}_p, \mathbb{Q}, \mathbb{R}$ or $\mathbb{Z}_q$ and for every $\beta > 1$ the β-gap-$\mathsf{SparseShift}_R$ problem is NP-hard. Furthermore, there exists a constant $\alpha > 1$ such that for every $d = O(1)$ the α^d-gap-$\mathsf{SparseShift}_R$ problem is NP-hard when given polynomials of degree at most d as input.*

The theorem is stated for the sparse representation model but as the polynomials under consideration have many non-zero terms it can also be stated without any modification in the dense representation model. We next state the theorem in the arithmetic circuit model.

▶ **Theorem 6** (Arithmetic circuit representation). *There exists a constant $\alpha > 1$ such that the following holds for $R = \mathbb{F}_p, \mathbb{Q}, \mathbb{R}$ or $\mathbb{Z}_q$. For every d the α^d-gap problem is NP-hard when given polynomials of degree at most d as input. Furthermore, our hard instances have circuit size $(nd + 1)$.*

Observe that if we take e.g. $d = n^c$ in the theorem above then the input size is $N = n^{c+1}$ and the gap is $\exp(N^{1-1/c})$.

2 Preliminaries

We use $[n]$ to refer to the set $\{1, 2, \ldots, n\}$. We use capital letters A and C to represent matrices, capital letters U, S and T to represent systems of equations, and capital letters X, Y and Z to represent sets of variables. We reserve letters x, y and z with, or without subscripts, to represent variables. We use bold letters $\mathbf{a}, \mathbf{b}, \ldots$ to indicate vectors and non-bold letters (apart from x, y and z) with, or without subscripts, $e, b_i, a_j, A_{i,j}, C_{k,\ell}, \ldots$ to indicate scalars.

Let S be a system of polynomial equations $\{f_i = 0\}_{i=1}^r$, where $\deg(f_i) = d_i$. We use $\mathrm{Vars}(f)$ to denote the variable support of the polynomial f, and for a system S of polynomial equations we use $\mathrm{Vars}(S)$ to denote the union of $\mathrm{Vars}(f_i)$ for all equations $f_i = 0$ in S. We denote with L an upper bound on the bit-complexity of the coefficients of the polynomials in the system.[3] The total degree of S is $d = \sum_i d_i$.

In this paper we shall consider two types of representations of polynomials (and hence of polynomial equations). The representation that is typically studied in the context of polynomial equations is the so called "sparse representation". In this representation polynomials are given as a set of pairs consisting of exponent vectors together with the coefficients of the corresponding monomials. E.g. the polynomial $2x^2z - y \in \mathbb{F}[x, y, z,]$ is represented as $\{((2, 0, 1), 2), (0, 1, 0), -1)\}$. This is called the sparse representation as we do not charge for monomials whose coefficients are equal to 0. In particular the size of the representation of a degree d polynomial can be much smaller than $\binom{n+d}{d}$. For a system of polynomial equations $S = \{f_i = 0\}_{i=1}^r$, the complexity of S, or its size, is defined to be the total bit size of the sparse representations of the polynomials $\{f_i\}_{i\in[r]}$. We note that this is always upper bounded by $\sum_{i=1}^r \binom{n+d_i}{d_i} \cdot L$.

The second type of representation that we consider is when the polynomials f_i are given as the outputs of arithmetic circuits (see [26, 24] for more details). In this paper we only consider the white-box version of this representation, i.e., when the computation graph of the circuit is explicitly given to the algorithm. In this case the complexity (or size) of the system S is the total size of the input circuits times the maximal bit complexity of coefficients in the circuits.

[3] When the underlying ring is an abstract ring one has to define this complexity, but for the usual rings and fields such as $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{F}_q$ this is the natural definition. In the BSS model this complexity is called height and is indeed only defined for these natural domains [2].

As we shall later see (Lemmata 7 and 8), given a system of equations, either via arithmetic circuits or in the sparse representation model, one can easily construct an equivalent system T of polynomial equations of degree 2 and roughly of the same complexity, such that the system S has a solution if and only if the system T does[4]. Hence, these two different representations have the same computational power. However, this reduction is not gap-preserving so we will have to give separate arguments for the gap problems.

3 Reduction from HN_R to $\mathsf{SparseShift}_R$

In this section, we shall first show that given a system S of r many polynomial equations over $R[X]$, we can algorithmically construct a system T of polynomial equations over $R[X'']$ such that $X \subseteq X''$; each polynomial in T is of degree at most 2; and if $\mathbf{a} \in R^{|X|}$ is a solution for the system S then there exists an extension $\mathbf{a}'$ of $\mathbf{a}$ such that $\mathbf{a}'$ is a solution for the system T. And vice versa, from a solution to T we deduce a solution to S. From T we shall then construct a polynomial $P_S \in R[X'', W]$ such that S has a solution if and only if the polynomial P_S can be sparsified.

3.1 Reduction to a system of polynomial equations of degree at most 2

Given a system S of polynomial equations, Lemma 7 and Lemma 8 summarize the construction of a system T which is as described in the paragraph above, for inputs given in sparse, and arithmetic circuit representations respectively.

▶ **Lemma 7** (Sparse representation). *Let $S = \{f_i(X) = 0\}_{i=1}^{r}$ be a system of polynomial equations over the polynomial ring $R[X]$ such that for each $i \in [r]$, $f_i(X)$ is a polynomial of degree d_i and sparsity s_i, and the bit complexity of each coefficient is at most L. Then, Algorithm 1 runs in time $\mathrm{poly}(|X|, r, \max_i\{d_i\}, \max_i\{s_i\}, L)$ and returns a set T of polynomial equations $\{g_j(X, Y, Z) = 0\}_{j=1}^{t}$ over the polynomial ring $R[X, Y, Z]$ such that*

- $t = \mathrm{poly}(|X|, r, \max_i\{d_i\}, \max_i\{s_i\}, L)$.
- $|X \sqcup Y \sqcup Z| = \mathrm{poly}(|X|, t, \max_i\{d_i\}, \max_i\{s_i\})$.
- *The bit-complexity of the coefficients of the polynomial equations in T is also at most L.*
- *Each polynomial equation $g_j(X, Y, Z) = 0$ in T is either a quadratic binomial polynomial equation or an affine linear polynomial equation.*
- *S has a solution $\mathbf{a} \in R^{|X|}$ if and only T has a solution $\mathbf{a}' \in R^{|X \sqcup Y \sqcup Z|}$.*

▶ **Lemma 8** (Arithmetic circuit representation). *Let $S = \{f_i(X) = 0\}_{i=1}^{r}$ be a system of polynomial equations over the polynomial ring $R[X]$ such that for each $i \in [r]$, the polynomial $f_i(X)$ is provided as an arithmetic circuit Φ_i of size s_i. Then, when given this as input, Algorithm 3 runs in time $\mathrm{poly}(|X|, r, \max_i\{s_i\})$ and returns a system T of polynomial equations $\{g_j(X, Y) = 0\}_{j=1}^{t}$ over the polynomial ring $R[X, Y]$ such that*

- $t = \mathrm{poly}(|X|, r, \max_i\{s_i\})$.
- $|Y| \leq r \cdot \max_i\{s_i\}$.
- *Each polynomial equation $g_j(X, Y) = 0$ in T is either a quadratic binomial polynomial equation or an affine linear polynomial equation.*
- *T has a solution in $R^{|X \sqcup Y|}$ if and only S has a solution in $R^{|X|}$.*

Due to the lack of space, we present the relevant details and proofs of Lemmata 7 and 8 in Appendix A.

[4] Such reductions were studied before. See [3, Proposition 1] and [5, Chapter 9].

3.2 Construction of P_S

Given a system S of polynomial equations over a set of variables X, in Section 3.1 we constructed the system T of polynomial equations of degree at most 2 over the set of variables X, Y and Z,[5] such that $|X \sqcup Y \sqcup Z|$ is at most polynomial in the input size. Without loss of generality, let the variables in $X \sqcup Y \sqcup Z$ be renamed as the variable set $X' = \{x_1, \ldots, x_N\}$ where $N = |X \sqcup Y \sqcup Z|$.

Let $\{g_1(X') = 0, \ldots, g_t(X') = 0\}$ be the enumeration of polynomial equations in T. Without loss of generality, we can assume that the number of equations with a non-zero constant term is equal to 1. Otherwise, given a system T of polynomial equations, with $t' > 1$ many of these polynomial equations with non-zero constant terms, we shall construct a new system T' such that the number of polynomial equations in T' that have non-zero constant terms is exactly equal to 1, and a solution of T is a solution of T' and vice versa. Without loss of generality assume that $\{g_1(X') = 0, \ldots, g_{t'}(X') = 0\}$ are the polynomial equations in T with non-zero constant terms. By Lemmata 7 and 8 it follows that $\{g_1(X') = 0, \ldots, g_{t'}(X') = 0\}$ are affine linear equations. Denote the free term in $g_1(X'), \ldots, g_{t'}(X')$ with $c_1, \ldots, c_{t'}$, respectively. We obtain T' from T by just updating each of the polynomials $g_i(X')$ (for $2 \leq i \leq t'$) as follows: $g_i(X') \leftarrow c_1 \cdot g_i(X') - c_i \cdot g_1(X')$. The rest of the polynomials from T are directly added to T'. It is easy to see that any solution to the system T is also a solution to the system T' and vice-versa (as R is an integral domain). Furthermore, the only equation in T' with a non-zero constant term is an affine linear equation.

Conditioned on the aforementioned discussion, we shall assume that all polynomial equations in T other than $g_1(X') = 0$, have no constant terms. For a new variable x_0, let $X'' = X' \sqcup \{x_0\}$ and thus $|X''| = N + 1$. Let $W = \{w_1, \ldots, w_t\}$ be a new set of variables disjoint from X''. Let γ be an element in R without a multiplicative inverse (recall that in Theorem 1 we assume that R is not a field). We shall now define our polynomial P_S in the polynomial ring $R[X'', W]$ as follows

$$P_S(X'', W) = \underbrace{w_1 \cdot g_1(X')}_{I} + \underbrace{\left(\sum_{i=2}^{t} w_i \cdot \left(\gamma \cdot g_i(X') + \sum_{k=0}^{N} x_k\right)\right)}_{II}. \tag{1}$$

Observe that $\deg(P_s) \leq 3$.

▶ Remark 9. The sparsity of the polynomial $P_S(X'', W)$, σ, is equal to the sum of sparsities of polynomials in each of its summands, and it is equal to $(t-1) \cdot (N+1) + \sum_{i=1}^{t} s'_i$ where s'_i is the sparsity of the polynomial $g_i(X')$. On the other hand, $P_S(X'', W)$ can also be represented as a depth four arithmetic circuit, with at most $3t$ non-leaf nodes.

We shall now show that it is sufficient to consider shifts with a certain structure for P_S. Further we shall show that a solution to the system S of polynomial equations exists if and only if there exists a vector $\mathbf{b}$ such that $P_S(X'' + \mathbf{b}, W)$ has fewer monomials than $P_S(X'', W)$.

▶ **Lemma 10.** *Let $\mathbf{a} = \{a_1, \ldots, a_N\} \in R^N$ be a solution to the system T of polynomial equations. Let $\mathbf{b} = \{b_0, \ldots, b_N\}, \mathbf{b}' = \{b'_0, \ldots, b'_N\} \in R^{N+1}$ and $\mathbf{a}' = \{a'_0, \ldots, a'_N\}$ be such that*

- *$b_i = b'_i$ for all $i \in [N]$,*
- *$b'_0 \neq b_0$ and $b_0 = -\sum_{i \in [N]} b_i$,*

[5] In case the polynomials in the system S of polynomial equations are provided as circuits, $Z = \emptyset$.

- $a'_i = a_i$ *for all* $i \in [N]$, *and*
- $a'_0 = -\sum_{i\in[N]} a_i$.

Let $\mathbf{b}''$ *and* $\mathbf{c}$ *be any vectors in* R^{N+1} *and* R^t *respectively. Then,*

1. *The sparsity of* $P_S(X'' + \mathbf{b}'', W + \mathbf{c})$ *is at least that of* $P_S(X'' + \mathbf{b}'', W)$,
2. *The sparsity of* $P_S(X'' + \mathbf{b}', W)$ *is at least that of* $P_S(X'' + \mathbf{b}, W)$.
3. *The sparsity of* $P_S(X'', W)$ *is* 1 *more than that of* $P_S(X'' + \mathbf{a}', W)$.

Proof. Given the structure of the polynomial $P_S(X'', W)$, proof of Item 1 follows directly from the fact that the polynomial $P_S(X'', W)$ is linear in the W variables and thus all terms of $P_S(X'' + \mathbf{b}'', W)$ also appear in $P_S(X'' + \mathbf{b}'', W + \mathbf{c})$. Further, the difference $P_S(X'' + \mathbf{b}'', W + \mathbf{c}) - P_S(X'' + \mathbf{b}'', W)$ does not depend on any W variable.

From their definition, the vectors $\mathbf{b}$ and $\mathbf{b}'$ are identical when projected down to their last N coordinates and these exactly correspond to shifts of variables in X'. We shall use $\mathbf{b}|_{X'}$ to denote this projection. In particular, for all $i \in [t]$, $g_i(X' + \mathbf{b}|_{X'}) = g_i(X' + \mathbf{b}'|_{X'})$. It is easy to see that the polynomial $\sum_{i=0}^{N} x_i$ is invariant under shift by $\mathbf{b}$ (as $\sum_{i=0}^{N} b_i = 0$) but not under shift by $\mathbf{b}'$. Putting both of these facts together we can now say that $P_S(X'' + \mathbf{b}', W)$ contains all the terms that are contained in $P_S(X'' + \mathbf{b}, W)$, and it additionally contains a non-trivial linear polynomial in the W variables. This proves Item 2 of the lemma.

Towards proving Item 3 of the lemma, we claim that under a shift by $\mathbf{a}'$, as defined in the statement of the lemma, sparsity of part II in Equation (1) does not change, and sparsity of part I definitely decreases.

All polynomial equations $\{g_i(X') = 0 \mid 2 \le i \le t\}$, are constant free and can either be quadratic binomial polynomial equations of the form $(x_p - x_q \cdot x_e) = 0$ (for some $p, q, e \in [N]$) or homogeneous linear polynomial equations of the form $\sum_{j=1}^{k} c_{i_j} x_{i_j} = 0$ (for some $i_1, \ldots, i_k \in [N]$ and scalars c_{i_j}).

When the equation is a quadratic binomial polynomial equation: Since $\mathbf{a} = \mathbf{a}'|_{X'}$ solves T we get that $a'_p - a'_q \cdot a'_e = 0$ and using this fact we can show that for each summand of this kind in part II, the sparsity does not change.

$$\begin{aligned}
&\gamma \cdot ((x_p + a'_p) - (x_q + a'_q) \cdot (x_e + a'_e)) + \sum_{i=0}^{N} (x_i + a'_i) \\
&= \gamma \cdot ((x_p - x_q \cdot x_e) - (a'_q x_e + a'_e x_q)) + \sum_{i=0}^{N} x_i \quad \text{(Since } a'_p - a'_q \cdot a'_e = 0 \text{ and } \sum_{i=0}^{N} a'_i = 0\text{)} \\
&= \gamma \cdot (x_p - x_q \cdot x_e) + \left(\sum_{i\in[N]\setminus\{q,e\}} x_i \right) + (1 - \gamma \cdot a'_q) \cdot x_e + (1 - \gamma \cdot a'_e) \cdot x_q.
\end{aligned}$$

Since γ has no multiplicative inverse, neither $(1 - \gamma \cdot a'_q)$ nor $(1 - \gamma \cdot a'_e)$ can be equal to 0.

When the polynomial equation is a homogeneous linear polynomial equation: Since $\mathbf{a} = \mathbf{a}'|_{X'}$ solves T we get that $\sum_{j=1}^{k} c_{i_j} a'_{i_j} = 0$ and thus the sparsity remains invariant for such summands in part II.

Finally consider the non-homogeneous linear polynomial equation $g_1(X') = 0$. Without loss of generality, let $g_1(X') = c_1 x_1 + \ldots + c_k x_k + c$. Note that $\sum_{i=1}^{k} c_i a'_i + c = 0$ as $\mathbf{a} = \mathbf{a}'|_{X'}$ solves T and thus sparsity reduces by 1 under shift by such a vector $\mathbf{a}'$:

$$\sum_{i=1}^{k} c_i (x_i + a'_i) + c = \sum_i c_i x_i + \left(\sum_{i=1}^{k} c_i a'_i + c \right) = \sum_{i=1}^{k} c_i x_i \ .$$

By putting together the analysis for all the summands we get that the polynomial $P_S(X'' + \mathbf{a}', W)$ has one monomial less than $P_S(X'', W)$. ◀

▶ **Lemma 11.** *Let* $\mathbf{b} = (b_0, b_1, \ldots, b_N) \in R^{N+1}$ *such that* $b_0 = -\sum_{i=1}^{N} b_i$*, be a shift that sparsifies the polynomial* $P_S(X'', W)$ *by at least one monomial. Let* $\mathbf{a} \in R^N$ *be the projection of vector* $\mathbf{b}$ *to its last* N *coordinates. Then* $\mathbf{a}$ *solves* T*.*

Proof. The proof of Lemma 10 shows that given the structure of the shift $\mathbf{b}$, a reduction in sparsity can only come from $g_1(X')$. That is, the sparsity of polynomials $g_i(X')$ for $i \geq 2$, can only increase upon a shift.

For the sake of contradiction, let us assume that there exists a polynomial equation in T that is not satisfied by $\mathbf{a}$. If for some $i \geq 2$, $g_i(X') = 0$ is a polynomial equation that is not satisfied by $\mathbf{a}$, then this contributes an increase of 1 to sparsity of the polynomial $P_S(X'', W)$ upon the shift by $\mathbf{b}$ (by adding a term of the form $c \cdot w_i$). Else if $g_1(X') = 0$ is not satisfied by $\mathbf{a}$, then there is no contribution to reduction in sparsity from part I. This is due to the fact that the term of the form $c \cdot w_1$ vanishes upon a shift by $\mathbf{b}$ if and only if $\mathbf{a}$ solves $g_1(X') = 0$. In either of these cases, the sparsity of $P_s(X'' + \mathbf{b}, W)$ is not strictly less than that of $P_S(X'', W)$. This contradicts our assumption that $\mathbf{b}$ sparsifies $P_S(X'', W)$ by at least one monomial. ◀

By putting together Lemmata 7, 8, 10, and 11, we get the following formal statement.

▶ **Theorem 12** (HN_R reduces to $\mathsf{SparseShift}_R$)**.** *Given a system* S *of polynomial equations over the polynomial ring* $R[X]$*, there exists a polynomial* $P_S(X'', W) \in R[X'', W]$ *(where* $X \subseteq X''$*) such that the system* S *is solvable if and only if there exists a shift that sparsifies the polynomial* P_S *by a monomial. Furthermore, the size of the polynomial instance* P_S *is polynomially related to the input size of the system* S *of polynomial equations. This holds true in both the sparse-representation and circuit-representation.*

We thus get that if $\mathsf{SparseShift}_R$ can be solved efficiently (in general) then HN_R can also be solved efficiently. In other words, $\mathsf{SparseShift}_R$ is at least as hard as HN_R. This completes the proof of Theorem 1. Putting Theorem 1 together with the fact that $\mathsf{HN}_{\mathbb{Z}}$ is undecidable (due to [22]).

4 Undecidability of β-gap-$\mathsf{SparseShift}_{\mathbb{Z}}$ problem

Note that $\mathsf{SparseShift}_R$ can be rephrased as the following gap problem – given a polynomial of sparsity σ, decide if there is a shift that sparsifies the polynomial to at most $\sigma - 1$ monomials, or there is no shift that sparsifies the polynomial below σ monomials. We shall now show a reduction from this ($\mathsf{SparseShift}_R$ problem) to β-gap-$\mathsf{SparseShift}_R$ for any $\beta > 1$.

Let the sets X'' and W be as defined in the construction of the polynomial P_S in Section 3.2. Let d be a parameter that we shall soon fix. Let $X^{(1)}, \ldots, X^{(d)}$ and $W^{(1)}, \ldots, W^{(d)}$ be d many disjoint copies of variable sets X'' and W respectively. Let $X_d = \sqcup_{k=1}^{d} X^{(i)}$ and $W_d = \sqcup_{k=1}^{d} W^{(i)}$. For the sake of brevity, let us use the following notation: Let $Y = X'' \sqcup W$. For all $k \in [d]$, let $Y^{(k)} = X^{(k)} \sqcup W^{(k)}$ and $\left|Y^{(k)}\right| = N'$. Let $Y_d = \sqcup_{k=1}^{d} Y^{(i)}$, so that $|Y_d| = N'd$. Let the polynomial $Q_d(Y_d)$ be defined as follows.

$$Q_d(Y_d) = \prod_{k=1}^{d} P_S(Y^{(k)}). \tag{2}$$

Observe that the sparsity of $Q_d(Y_d)$ is given by the product of sparsities of d many instances of $P_S(X'', W)$.

▶ **Lemma 13.** *Let $P_S(X'', W)$ and $Q_d(Y_d)$ be the polynomials as defined above. Let σ be equal to the sparsity of the polynomial P_S. Then,*

1. $\deg(Q_d) \leq 3d$.
2. *$Q_d(Y_d)$ has sparsity equal to σ^d.*
3. *If P_S has a depth four circuit of size $s \leq 3t + N'$ (recall Remark 9) then Q_d has a depth five circuit of size $sd + 1$.*
4. *There is a vector $\mathbf{a} \in R^{N'}$ such that $P_S(Y + \mathbf{a})$ has at most $\sigma - 1$ monomials if and only if there exists a vector $\mathbf{a}_d \in R^{N' \cdot d}$ such that $Q_d(Y_d + \mathbf{a}_d)$ has at most $(\sigma - 1)^d$ monomials.*
5. *For all vectors $\mathbf{a} \in R^{N'}$, $P_S(Y + \mathbf{a})$ has at least σ monomials if and only if for all vectors $\mathbf{a}_d \in R^{N' \cdot d}$ $Q_d(Y_d + \mathbf{a}_d)$ has at least σ^d monomials.*

Proof. The claim regarding the degree of Q_d follows immediately from the fact that $\deg(P_S) \leq 3$. Given that $P_S(X'', W)$ has a sparsity of σ and since $Q_d(Y_d)$ is defined to be a product of d distinct copies of $P_S(X'', W)$, sparsity of $Q_d(Y_d)$ is equal to σ^d. Similarly, if P_S can be computed by a circuit of size s, then there is a depth five circuit of size $(sd + 1)$ that computes the polynomial $Q_d(Y_d)$ – its output node is a product node into which d copies of circuits of $P_S(X'', W)$ feed into.

If there is a vector $\mathbf{a} \in R^{N'}$ such that $P_S(Y + \mathbf{a})$ has at most $\sigma - 1$ monomials then by taking $\mathbf{a}_d$ to be the concatenation of $\mathbf{a}$, d many times, we get that $Q_d(Y_d + \mathbf{a}_d)$ has at most $(\sigma - 1)^d$ monomials. If there is $\mathbf{a}_d \in R^{N' \cdot d}$ such that $Q_d(Y_d + \mathbf{a}_d)$ has at most $(\sigma - 1)^d$ monomials then it cannot happen that there is no $\mathbf{a} \in R^{N'}$ such that $P_S(Y + \mathbf{a})$ has at most $\sigma - 1$ monomials.

If for all vectors $\mathbf{a} \in R^{N'}$, $P_S(Y + \mathbf{a})$ has at least σ monomials, then $Q_d(Y_d + \mathbf{a}_d)$ must have at least σ^d monomials for all $\mathbf{a}_d \in R^{N' \cdot d}$. On the other hand if for all vectors $\mathbf{a}_d \in R^{N' \cdot d}$, $Q_d(Y_d + \mathbf{a}_d)$ has at least σ^d monomials, (for the sake of contradiction) let us suppose that there is a vector $\mathbf{a}' \in R^{N'}$ such that $P_S(Y + \mathbf{a}')$ has at most $\sigma - 1$ monomials. As before (due to the product structure of Q_d) we get that there is a corresponding vector $\mathbf{a}'_d \in R^{N' \cdot d}$ such that $Q_d(Y_d + \mathbf{a}'_d)$ has at most $(\sigma - 1)^d$ monomials. This contradicts our assumption. Thus, for all $\mathbf{a} \in R^{N'}$, $P_S(Y + \mathbf{a})$ has at least σ monomials. ◀

▶ **Theorem 14.** *Let R be an integral domain but not a field. Given a system S of polynomial equations over the polynomial ring in n variables $R[X]$, for any function $\beta : \mathbb{N} \to \mathbb{R}_+$ such that $\beta \in o(1)$, there exist $d = d(S, \beta) \in \mathbb{Z}_{>0}$ and a polynomial $Q_d(Y_d)$, in $N'd$ variables, of degree at most $3d$, such that the system S is solvable if and only if the M^β-gap-$\mathsf{SparseShift}_R$ problem for $Q_d(Y_d)$ is solvable, where M is the representation length of $Q_d(Y_d)$ in the sparse representation.*[6]

Proof. Given a system S of polynomial equations, we can construct the polynomial $P_S(X'', W)$ (as defined in Equation (1)). Let σ be the sparsity of $P_S(X'', W)$. Recall from Theorem 12 that system S has a solution if and only if there exists a shift that sparsifies the polynomial $P_S(X'', W)$ by a monomial.

Recall that $Q_d(Y_d)$ has $M' = \sigma^d$ many monomials. Let $\alpha = \frac{\sigma}{\sigma - 1}$. By putting together Theorem 12 and Lemma 13, we get that the system S of polynomial equations is solvable if and only if the α^d-gap problem for $Q_d(Y_d)$ is solvable. Calculating we get that $\alpha^d = \left(\frac{\sigma}{\sigma-1}\right)^d \approx e^{d/(\sigma-1)} = M'^{\frac{1}{(\sigma-1)\log\sigma}}$. Picking d large enough so that $M' \geq \sqrt{M}$ and $\beta(M) < \frac{1}{2(\sigma-1)\log\sigma}$ the claim follows. ◀

[6] Recall that in sparse representation polynomials are given as a set of pairs consisting of exponent vectors together with the coefficient of the corresponding monomial. Thus, $M \approx (\#\text{ monomials in } Q_d) \times ((N'd) \times O(\log d)) \times O(d \cdot b)$, where b is the maximal bit complexity of a coefficient in $P_S(X'', W)$.

Putting Theorem 14 together with the fact that $\mathsf{HN}_{\mathbb{Z}}$ is undecidable (due to [22]) we get Theorem 4.

5 Hardness of β-gap-SparseShift$_R$ problem for $R = \mathbb{F}_p, \mathbb{Q}, \mathbb{R}$, or $\mathbb{Z}_q$

In this section we prove Theorems 5 and 6 by giving a reduction from Max-3Lin$_R$ to the α-gap-SparseShift$_R$ problem, for different domains $R = \mathbb{F}_p, \mathbb{Q}, \mathbb{R}$ or $\mathbb{Z}_q$. Observe that we now do not require that our ring R is not a field.

Let $X = \{x_1, \ldots, x_n\}$. Let the given system S of linear equations be $\{L_1(X) = 0, \ldots, L_m(X) = 0\}$, where $L_i(X) \in R[X]$, and each equation $L_i(X) = 0$ depends on exactly 3 variables. Let the given system of equations be expressed together as $A \cdot X + \mathbf{b} = \mathbf{0}$ such that for all $i \in [m]$, $A_i \cdot X + b_i = 0$ is the i'th linear equation $L_i(X) = 0$, where A_i is the i'th row of the matrix A. Note that there are exactly three non-zero entries in each row of A. Let $w = \max\{2n, 2m\}$. Let C be a $w \times w$ matrix such that

$$\text{for all } 1 \le i, j \le w, \quad C_{i,j} = \begin{cases} A_{i,j-w+n} & \text{if } i \le m \text{ and } j \ge w - n + 1; \\ 0 & \text{otherwise.} \end{cases}$$

In other words, A is the top right block of C and the rest of C is zeros. Let $\mathbf{e} = (e_1, \ldots, e_w) \in R^w$ be such that $e_i = b_i$ for all $1 \le i \le m$ and $e_i = 0$ otherwise. Let e_0 be some constant. Let $Y = \{y_1, \ldots, y_w\}$ be a new set of variables disjoint from X. Let the polynomial $Q_S(Y) \in R[Y]$ be defined as follows.

$$Q_S(Y) = \sum_{i,j \in [w]} C_{i,j} \cdot y_i y_j + \sum_{i \in [w]} e_i \cdot y_i + e_0 \,.$$

Note that there are at most $3m$ many non-zero entries in C, and there are at most m many non-constant linear terms. Thus the sparsity of this polynomial is at most $4m + 1$.

For some vector $\mathbf{a} = (a_1, \ldots, a_w) \in R^w$ let us examine the structure of the polynomial $Q_S(Y + \mathbf{a})$.

$$\begin{aligned} Q_S(Y + \mathbf{a}) &= \sum_{i,j \in [w]} C_{i,j} \cdot (y_i + a_i)(y_j + a_j) + \sum_{i \in [w]} e_i \cdot (y_i + a_i) + e_0 \\ &= \sum_{i,j \in [w]} C_{i,j} \cdot (y_i y_j + a_i y_j + a_j y_i + a_i a_j) + \sum_{i \in [w]} e_i \cdot (y_i + a_i) + e_0 \\ &= \sum_{i,j \in [w]} C_{i,j} \cdot (y_i y_j + a_i a_j) + \sum_{i,j \in [w]} a_i \cdot y_j \cdot C_{i,j} + \sum_{i,j \in [w]} a_j \cdot y_i \cdot C_{i,j} + \sum_{i \in [w]} e_i \cdot (y_i + a_i) + e_0 \\ &= \sum_{i,j \in [w]} C_{i,j} \cdot y_i y_j + \sum_{i \in [w]} y_i \cdot (e_i + \sum_{j \in [w]} a_j \cdot (C_{i,j} + C_{j,i})) + \sum_{i,j \in [w]} C_{i,j} \cdot a_i a_j + \sum_{i \in [w]} a_i \cdot e_i + e_0 \,. \end{aligned}$$

Observe that the quadratic part of the polynomial $Q_S(Y)$ remains unperturbed under the shift but the affine linear part of it could get perturbed. We shall now show that every non-zero coefficient in the linear part corresponds to a linear equation in S.

▶ **Lemma 15.** *Let $\mathbf{a} = (a_1, \ldots, a_w) \in R^w$. Let $\mathbf{a}' \in R^n$ be the projection of $\mathbf{a}$ down to its last n elements, that is, for all $j \in [n]$, $a'_j = a_{j+w-n}$. Then, for all $i \in [m]$, the coefficient of y_i in $Q_S(Y + \mathbf{a})$ is zero if and only if $\mathbf{a}'$ satisfies the i'th linear equation $L_i(X) = 0$. Moreover, for all $i > m$, the coefficient of y_i is zero.*

Proof. For all $i \in [m]$, the coefficient of y_i in the polynomial $Q_S(Y + \mathbf{a})$ is equal to

$$e_i + \sum_{j \in [w]} a_j (C_{i,j} + C_{j,i}) \,.$$

Note that for this regime of $i \in [m]$, $e_i = b_i$, $C_{j,i} = 0$, and $C_{i,j} = A_{i,j-w+n}$ for j in $[w-n+1, w]$ and zero otherwise. Thus, the coefficient of such a y_i in $Q_S(Y + \mathbf{a})$ reduces as follows.

$$e_i + \sum_{j\in[w]} a_j(C_{i,j} + C_{j,i}) = b_i + \sum_{j'=1}^{n} a_{w-n+j'} A_{i,j'} = b_i + \sum_{j'=1}^{n} a'_{j'} A_{i,j'}.$$

This is exactly the value obtained by evaluating the i'th linear polynomial L_i at $\mathbf{a}'$. Thus, we get that coefficient of y_i (for $i \in [m]$) is zero if and only if $\mathbf{a}'$ satisfies the i'th linear equation, i.e., $L_i(\mathbf{a}') = 0$.

For all $i > m$, $e_i = 0$ and $C_{i,j} = 0$. Further, $\mathbf{a}_j = 0$ for all $j \leq w - n$. Hence,

$$e_i + \sum_{j\in[w]} a_j(C_{i,j} + C_{j,i}) = \sum_{j\in[w]} a_j \cdot C_{j,i} = \sum_{j=w-n+1}^{w} a_j \cdot C_{j,i} = 0\,.$$

The last equality in the math block above is due to the fact that the entries $C_{j,i}$ are equal to zero for $j \geq w - n + 1$ and $i \geq m + 1$ regardless of whether $n \geq m$ or $m \geq n$, from the construction of the matrix C. Thus the coefficients of the terms y_i for all $i > m$ are zero. ◀

Using this correspondence, we can show the following reduction.

▶ **Lemma 16.** *Let $\mathbf{a} = (a_1, \ldots, a_w) \in R^w$. Let $\mathbf{a}' \in R^n$ be the projection of $\mathbf{a}$ down to its last n elements, that is, for all $j \in [n]$, $a'_j = a_{j+w-n}$. Then*

1. *$\mathbf{a}'$ satisfies at most δ fraction of equations in S if and only if $Q_S(Y + \mathbf{a})$ has at least $(4-\delta)m$ non-constant monomials, and*
2. *$\mathbf{a}'$ satisfies at least $1-\varepsilon$ fraction of equations in S if and only if $Q_S(Y + \mathbf{a})$ has at most $(3+\varepsilon)m + 1$ monomials.*

Proof. From the aforementioned discussion, the sparsity of the polynomial $Q_S(Y + \mathbf{a})$ is decided by the coefficients of the linear terms. Further, Lemma 15 characterizes that the coefficient of a linear term is zero if and only if the *corresponding* linear polynomial equation is *satisfied*. Thus at most δ fraction of equations in S are satisfied if and only if at most δ fraction of coefficients of linear terms are equal to zero. In other words, if at least $(1-\delta)$ fraction of coefficients of linear terms are non-zero. The constant term e_0 could get cancelled out, that is, $\sum_{i,j\in[w]} C_{i,j} \cdot a_i a_j + \sum_{i\in[w]} a_i \cdot e_i + e_0$ could be zero. Thus, $\mathbf{a}'$ satisfies at most δ fraction of equations in S if and only if $Q_S(Y + \mathbf{a})$ has at least $3m + (1-\delta)m = (4-\delta)m$ non-constant monomials.

Similarly at least $1-\varepsilon$ fraction of equations in S are satisfied if and only if at least $1-\varepsilon$ fraction of coefficients of linear terms are equal to zero. In other words, if at most ε fraction of coefficients of linear terms are non-zero. Thus, $\mathbf{a}'$ satisfies at least $1-\varepsilon$ fraction of equations in S if and only if $Q_S(Y + \mathbf{a})$ has at most $(3+\varepsilon)m + 1$ monomials. ◀

We first prove a more restricted version of Theorems 5 and 6 that shows hardness of α-approximate $\mathsf{SparseShift}_R$ for some small α. Then we shall amplify this hardness for any $\beta > 1$.

▶ **Theorem 17.** *Let $R = \mathbb{F}_p, \mathbb{R}, \mathbb{Q}$ or $\mathbb{Z}_q$. For all $\varepsilon, \delta > 0$ as given in the last column of Table 1, there exists an $\alpha = \alpha(\varepsilon, \delta, R)$ such that it is* NP*-hard to α-approximate $\mathsf{SparseShift}_R$, in either the sparse, dense or arithmetic circuit representation.*

Proof. We first note that as the input is a degree 2 polynomial, all three representations are polynomially equivalent.

Suppose for a regime of values of $\varepsilon', \delta' > 0$ we are guaranteed the following. Given an instance of Max-3Lin$_R$, it is NP-hard to distinguish the following cases – if there is a assignment that satisfies at least $(1-\varepsilon')$ fraction of linear equations or for all assignments at most δ' fraction of linear equations are satisfied. Putting this together with Lemma 16, we get that it is NP-hard to distinguish if there is a vector $\mathbf{a}$ such that the polynomial $Q_S(Y+\mathbf{a})$ has at most $t = 3m+\varepsilon' m+1$ monomials or if for all $\mathbf{a}$, the polynomial $Q_S(Y+\mathbf{a})$ has at least $\alpha t = (4-\delta')m$ non-constant monomials. Thus, we get that it is NP-hard to α-approximate SparseShift$_R$ where $\alpha = \alpha(\varepsilon', \delta')$ is obtained as follows.

$$\alpha = \frac{4-\delta'}{3+\varepsilon'+\frac{1}{m}} = \frac{4}{3} - \frac{4\varepsilon' + 3\delta' + o(1)}{9 + 3\varepsilon' + o(1)}.$$

Each row of Table 1 gives us a guarantee of the form that we assumed at the beginning of this proof. Thus by iterating through the rows of Table 1, we get our parameter $\alpha = \alpha(\varepsilon, \delta, R)$ for various settings of R. This completes the proof. ◀

Let d be a parameter that we shall soon fix. Let $Y^{(1)}, \ldots, Y^{(d)}$ be d many disjoint copies of the variable set $Y = \{y_1, \ldots, y_w\}$. Let $Y_d = \sqcup_{k=1}^{d} Y^{(i)}$. Let the polynomial $F_{n,d}(Y_d)$ be defined as follows.

$$F_{n,d}(Y_d) = \prod_{k=1}^{d} Q_S(Y^{(k)}). \tag{3}$$

Observe that the sparsity of $F_{n,d}(Y_d)$ is given by the product of sparsities of d many instances of $Q_S(Y)$. Further, if the polynomial $Q_S(Y)$ is computed by a circuit of size s then the polynomial $F_{n,d}(Y_d)$ has a circuit of size at most $sd+1$.

▶ **Lemma 18.** *Let S be a system of linear equations, and $F_{n,d}(Y_d)$ be the polynomial as defined above.*

1. *All vectors $\mathbf{a} \in R^n$ satisfy at most δ fraction of equations in S if and only if all vectors $\mathbf{b}_d \in R^{wd}$ are such that $F_{n,d}(Y_d + \mathbf{b}_d)$ has at least $((4-\delta)m)^d$ non-constant monomials.*
2. *There exists a vector $\mathbf{a} \in R^n$ such that it satisfies at least $1-\varepsilon$ fraction of equations in S if and only if there exists a vector $\mathbf{b}_d \in R^{wd}$ such that $F_{n,d}(Y_d + \mathbf{b}_d)$ has at most $((3+\varepsilon)m+1)^d$ monomials.*

Proof. From the product structure of $F_{n,d}(Y_d)$, we get that for all vectors $\mathbf{b}_d \in R^{wd}$, the polynomial $F_{n,d}(Y_d + \mathbf{b}_d)$ has at least $((4-\delta)m)^d$ non-constant monomials if and only if for all vectors $\mathbf{a} \in R^n$, the polynomial $Q_S(Y+\mathbf{a})$ has at least $(4-\delta)m$ non-constant monomials. For the sake of contradiction, let us suppose that there is a vector $\mathbf{b}'_d \in R^{wd}$ such that $F_{n,d}(Y_d + \mathbf{b}'_d)$ has at most $((4-\delta)m)^d - 1$ monomials. Because of the product structure of $F_{n,d}(Y_d)$, it must the case that there is a copy of $Q_S(Y)$, say $Q_S(Y^{(i)})$ such that $Q_S(Y^{(i)} + \mathbf{b}'_d|_{Y^{(i)}})$ has at most $(4-\delta)m - 1$ non-constant monomials which contradicts our assumption. The other direction also follows trivially from the product structure. From Lemma 16, we get that $\mathbf{a}'$ satisfies at most δ fraction of equations in S if and only if $Q_S(Y+\mathbf{a})$ has at least $(4-\delta)m$ non-constant monomials. By putting both of these together, we get Item 1.

By invoking Lemma 16 again, we get that there is a vector $\mathbf{a} \in R^n$ such that it satisfies at least $1-\varepsilon$ fraction of equations in S if and only if there is a vector $\mathbf{b}$ such that $Q_S(Y+\mathbf{b})$ has at most $(3+\varepsilon)m+1$ monomials. By taking $\mathbf{b}_d$ to be the concatenation of $\mathbf{b}$, d many times,

we get that $F_{n,d}(Y_d + \mathbf{b}_d)$ has at most $((3+\varepsilon)m+1)^d$ monomials. On the other hand, if there exists a vector $\mathbf{b}_d \in R^{wd}$ such that $F_{n,d}(Y_d + \mathbf{b}_d)$ has at most $((3+\varepsilon)m+1)^d$ monomials then because of the product structure of $F_{n,d}(Y_d)$, $Q_S(Y + \mathbf{a})$ has at most $(3+\varepsilon)m+1$ monomials where $\mathbf{a} = \mathbf{b}_d|_{Y^{(1)}}$. This completes the proof of Item 2. ◀

Proof of Theorems 5 and 6. Given any $\beta > 1$ and α as given by Theorem 17, let $d = \log_\alpha \beta$. Thus, $\beta = \alpha^d$. Let $F_{n,d}(Y_d)$ be the polynomial as defined in Equation (3). From Lemma 18, we get that α-gap-$\mathsf{SparseShift}_R$ gap reduces to α^d-gap-$\mathsf{SparseShift}_R$.

To prove Theorem 5 we note that if $Q_S(Y)$ is provided in sparse representation (recall that sparsity of $Q_S(Y)$, denoted by t, is at most $4m+1$) and if α^d-gap-$\mathsf{SparseShift}_R$ problem for $F_{n,d}(Y_d)$ can be solved efficiently in time $N^{O(1)}$ (where $N = t^d$ is the sparsity of the polynomial $F_{n,d}(Y_d)$) then α-gap-$\mathsf{SparseShift}_R$ problem for $Q_S(Y)$ can be solved in time $t^{O(d)}$. Thus, as long as $d = O(1)$ the gap reduction runs in polynomial time.

Similarly, to prove Theorem 6 we note that if $Q_S(Y)$ is provided as a circuit of size s and if α^d-gap-$\mathsf{SparseShift}_R$ problem for $F_{n,d}(Y_d)$ can be solved efficiently in time $N^{O(1)}$ (where $N = sd+1$ is the input size of the instance provided as a circuit) then α-gap-$\mathsf{SparseShift}_R$ problem $Q_S(Y)$ can be solved in time $(sd)^{O(1)}$. As long as d is at most a polynomial in s, the gap reduction runs in polynomial time. ◀

References

1. Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynominal interpolation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 301–309. ACM, 1988. doi:10.1145/62212.62241.
2. Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale. *Complexity and real computation*. Springer, 1998. URL: https://link.springer.com/book/10.1007/978-1-4612-0701-6.
3. Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP- completeness, recursive functions and universal machines. *Bulletin (New Series) of the American Mathematical Society*, 21(1):1–46, 1989. doi:bams/1183555121.
4. Allan Borodin and Prasoon Tiwari. On the decidability of sparse univariate polynomial interpolation. *Comput. Complex.*, 1:67–90, 1991. doi:10.1007/BF01200058.
5. Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity and beyond. *Foundations and Trends® in Theoretical Computer Science*, 6(1–2):1–138, 2011. doi:10.1561/0400000043.
6. Michael Clausen, Andreas W. M. Dress, Johannes Grabmeier, and Marek Karpinski. On zero-testing and interpolation of k-sparse multivariate polynomials over finite fields. *Theor. Comput. Sci.*, 84(2):151–164, 1991. doi:10.1016/0304-3975(91)90157-W.
7. Martin Davis. Hilbert's tenth problem is unsolvable. *The American Mathematical Monthly*, 80(3):233–269, 1973. doi:10.1080/00029890.1973.11993265.
8. Zeev Dvir, Rafael Mendes de Oliveira, and Amir Shpilka. Testing equivalence of polynomials under shifts. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *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. doi:10.1007/978-3-662-43948-7_35.
9. Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 563–574. IEEE Computer Society, 2006. doi:10.1109/FOCS.2006.51.
10. Dima Grigoriev. Testing shift-equivalence of polynomials by deterministic, probabilistic and quantum machines. *Theor. Comput. Sci.*, 180(1-2):217–228, 1997. doi:10.1016/S0304-3975(96)00188-0.

11 Dima Grigoriev and Marek Karpinski. Algorithms for sparse rational interpolation. In Stephen M. Watt, editor, *Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation, ISSAC '91, Bonn, Germany, July 15-17, 1991*, pages 7–13. ACM, 1991. doi:10.1145/120694.120696.

12 Dima Grigoriev and Marek Karpinski. A zero-test and an interpolation algorithm for the shifted sparse polynominals. In Gérard D. Cohen, Teo Mora, and Oscar Moreno, editors, *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes, 10th International Symposium, AAECC-10, San Juan de Puerto Rico, Puerto Rico, May 10-14, 1993, Proceedings*, volume 673 of *Lecture Notes in Computer Science*, pages 162–169. Springer, 1993. doi:10.1007/3-540-56686-4_41.

13 Dima Grigoriev, Marek Karpinski, and Michael F. Singer. Interpolation of sparse rational functions without knowing bounds on exponents. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume II*, pages 840–846. IEEE Computer Society, 1990. doi:10.1109/FSCS.1990.89616.

14 Dima Grigoriev and Yagati N. Lakshman. Algorithms for computing sparse shifts for multivariate polynomials. *Appl. Algebra Eng. Commun. Comput.*, 11(1):43–67, 2000. doi:10.1007/s002000050004.

15 Venkatesan Guruswami and Prasad Raghavendra. Hardness of learning halfspaces with noise. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 543–552. IEEE Computer Society, 2006. doi:10.1109/FOCS.2006.33.

16 Venkatesan Guruswami and Prasad Raghavendra. A 3-query PCP over integers. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 198–206. ACM, 2007. doi:10.1145/1250790.1250819.

17 Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. doi:10.1145/502090.502098.

18 Erich Kaltofen, Yagati N. Lakshman, and J.-M. Wiley. Modular rational sparse multivariate polynomial interpolation. In Shunro Watanabe and Morio Nagata, editors, *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC '90, Tokyo, Japan, August 20-24, 1990*, pages 135–139. ACM, 1990. doi:10.1145/96877.96912.

19 Neeraj Kayal. Affine projections of polynomials: extended abstract. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19–22, 2012*, pages 643–662. ACM, 2012. doi:10.1145/2213977.2214036.

20 Pascal Koiran. Hilbert's nullstellensatz is in the polynomial hierarchy. *J. Complex.*, 12(4):273–286, 1996. doi:10.1006/jcom.1996.0019.

21 Yagati N. Lakshman and B. David Saunders. Sparse polynomial interpolation in nonstandard bases. *SIAM J. Comput.*, 24(2):387–397, 1995. doi:10.1137/S0097539792237784.

22 Yuri V. Matiyasevich. The diophantineness of enumerable sets. *Dokl. Akad. Nauk SSSR*, 191(2):279–283, 1970. URL: http://mi.mathnet.ru/dan35274.

23 Dori Medini and Amir Shpilka. Hitting sets and reconstruction for dense orbits in vp_{e} and ΣΠΣ circuits. In Valentine Kabanets, editor, *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. doi:10.4230/LIPIcs.CCC.2021.19.

24 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity, version 9.0.3. Github survey, 2021. URL: https://github.com/dasarpmar/lowerbounds-survey/releases/tag/v9.0.3.

25 Shubhangi Saraf and Sergey Yekhanin. Noisy interpolation of sparse polynomials, and applications. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, USA, June 8-10, 2011*, pages 86–92. IEEE Computer Society, 2011. doi:10.1109/CCC.2011.38.

26 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388, 2010. doi:10.1561/0400000039.

27 Leslie G. Valiant. Completeness classes in algebra. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 – May 2, 1979, Atlanta, Georgia, USA*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

A Reduction to a system of polynomial equations of degree at most 2

Case when input is provided in sparse representation

Let $S = \{f_1 = 0, \ldots, f_r = 0\}$ be our input system of polynomial equations such that each f_i is provided in sparse representation. Let s_i denote the number of monomials with non-zero coefficients in the polynomial f_i. Let Y and Z be new disjoint sets of variables, that are disjoint from X such that $Y = \{y_{j,k}^{(i)} \mid i \in [r], j \in [s_i] \text{ and } k \geq 1\}$ and $Z = \{z_j^{(i)} \mid i \in [r], j \in [s_i]\}$.

Let the variables in Y have a lexicographic ordering based on the indices i, j and k, variables in Z have a lexicographic ordering based on the indices i and j, and the variables in X have some arbitrary ordering. Across the sets X, Y and Z, let the ordering be $Z \succ Y \succ X$. Given $S = \{f_1 = 0, \ldots, f_r = 0\}$, we construct an extended system T of polynomial equations over the variables $X \sqcup Y \sqcup Z$ such that each polynomial equation is of degree at most 2 and is such that there is a solution $\mathbf{a} \in R^{|X|}$ for the system S if and only if there exists an extension $\mathbf{a}'$ of $\mathbf{a}$ such that $\mathbf{a}'$ is a solution for the system T. This is a well known reduction (see, e.g., Lemma 6 in Chapter 2 of [2]) but for completeness we repeat it here.

Algorithm 1 and Algorithm 2 describe the construction of the extended system of equations. What the algorithms do is, roughly, for any monomial $m = x_{i_1} \cdot x_{i_2} \cdot x_{i_3} \cdot \ldots \cdot x_{i_j}$ of degree greater than 2, introduce a new variable, say y, replace m with the the monomial $y \cdot x_{i_3} \cdot \ldots \cdot x_{i_j}$ and introduce a new equation $y - x_{i_1} \cdot x_{i_2} = 0$ and for any monomial m', of degree 1, introduce a new variable z and a new equation $z - m' = 0$. Finally, an affine linear equation of the form $\sum c_i z_i + c_0$ is added to account for the fact that the original sum of monomials has to be zero. In particular, at the termination of the algorithm, the system T consists of constant-free quadratic binomial[7] equations and affine linear polynomial equations. It is also clear that there exists $\mathbf{a}'$ that satisfies T if and only if there is $\mathbf{a}$ that satisfies S.

▶ **Lemma 19** (Lemma 7 restated). *Let $S = \{f_i(X) = 0\}_{i=1}^r$ be a system of polynomial equations over the polynomial ring $R[X]$ such that for each $i \in [r]$, $f_i(X)$ is a polynomial of degree d_i and sparsity s_i, and the bit complexity of each coefficient is at most L. Then, Algorithm 1 runs in time $\mathrm{poly}(|X|, r, \max_i\{d_i\}, \max_i\{s_i\}, L)$ and returns a set T of polynomial equations $\{g_j(X, Y, Z) = 0\}_{j=1}^t$ over the polynomial ring $R[X, Y, Z]$ such that*

- $t = \mathrm{poly}(|X|, r, \max_i\{d_i\}, \max_i\{s_i\}, L)$.
- $|X \sqcup Y \sqcup Z| = \mathrm{poly}(|X|, t, \max_i\{d_i\}, \max_i\{s_i\})$.
- *The bit-complexity of the coefficients of the polynomial equations in T is also at most L.*
- *Each polynomial equation $g_j(X, Y, Z) = 0$ in T is either a quadratic binomial polynomial equation or an affine linear polynomial equation.*
- *S has a solution $\mathbf{a} \in R^{|X|}$ if and only T has a solution $\mathbf{a}' \in R^{|X \sqcup Y \sqcup Z|}$.*

Proof. Note that by the aforementioned ordering of variables, in all quadratic binomial equations included into the set T that have the form $u - v \cdot w = 0$, we have that u is of the form $y_{j,k}^{(i)}$, and v and w could be of the form $y_{j,k'}^{(i)}$ or $x_{i'}$, and the term u is leading with respect

[7] We use the phrase constant-free quadratic binomial equation to refer to an equation with two non-constant monomials of degree at most 2.

Algorithm 1 ConstructExtendedSystem-SparseRepresentation(S).

Result: Given a system $S = \{f_1 = 0, \ldots, f_r = 0\}$ of polynomial equations provided in sparse representation, we construct a system T of polynomial equations over an extended set of variables such that each polynomial in T has degree at most 2 and if $\mathbf{a}$ solves S then there exists an extension $\mathbf{a}'$ of $\mathbf{a}$ such that $\mathbf{a}'$ solves T and vice-versa.

$T \leftarrow \emptyset$;
for $i \in [r]$ **do**
  $f' \leftarrow 0$;
  Let s_i be the sparsity of f_i;
  for $j \in [s_i]$ **do**
    Let $c_{i,j}$ be the coefficient of j'th monomial $m_{i,j}$ in f_i;
    if $\deg(m_{i,j}) \geq 1$ **then**
      $U \leftarrow \text{ReduceMonomial}(m_{i,j}, i, j)$;
      $T \leftarrow T \cup U$;
      $f' \leftarrow f' + c_{i,j} \cdot z_j^{(i)}$;
    else
      $f' \leftarrow f' + c_{i,j}$;
    end
  end
  $T \leftarrow T \cup \{f' = 0\}$;
end
return T

to the terms v and w. Further, all the quadratic equations in the set T can be assumed to have some sort of a *topological order*. Thus the values of all the Y variables that appear in the variable support can be inductively inferred by just setting the X variables. That is, if we want to satisfy all such equations, then the value of the term u can be inferred from the value of terms v and w for every invocation of u, v and w.

Further note that some of the linear polynomial equations in T take the form $u' - v' = 0$ (from Algorithm 2 of Algorithm 2) where u' is of the form $z_j^{(i)}$, and v' could be of the form $y_{j,k}^{(i)}$ or $x_{i'}$. Similar to the case above, the value of the term u' can be inferred from the value of the term v' for every invocation of u' and v', and the value of v' is already fixed as the values of all the X and Y variables that appear in the variable support were set in the aforementioned discussion. Observe that the rest of the linear polynomial equations in T (from Algorithm 1 of Algorithm 1) correspond to the polynomial equations in S, and setting of Z variables in the variable support of T, by the above procedure, satisfies all the linear polynomial equations in T.

Given an assignment $\mathbf{a}$ to X let $\mathbf{a}'$ be its unique extension to the variable set $X \sqcup Y \sqcup Z$ according to the process described above. The argument above shows that $\mathbf{a}$ is a solution to the system S if and only if $\mathbf{a}'$ is a solution to the system T. ◀

Case when input is provided in white-box circuit form

Let $S = \{f_i = 0\}_{i=1}^{r}$ be our input system of polynomial equations such that each f_i (for $i \in [r]$) is provided as an arithmetic circuit Φ_i of size s_i. Without loss of generality, for all $i \in [r]$, we can assume that every product gate in Φ_i has a fan-in of 2.

Algorithm 2 ReduceMonomial(m, i, j).

Result: Given the j'th monomial (of degree at least 1) of the i'th polynomial, $m_{i,j}$, generate a collection of polynomial equations U from it.

$U \leftarrow \emptyset$;
$k = 1$;
while $\deg(m) \geq 2$ **do**
  Let u, v be the renaming of the two trailing variables under the ordering of X and Y variables as described above;
  $U \leftarrow U \cup \{y_{j,k}^{(i)} - u \cdot v = 0\}$;
  $m \leftarrow \frac{m}{u \cdot v} \cdot y_{j,k}^{(i)}$;
  $k \leftarrow k + 1$;
end
$U \leftarrow U \cup \{z_j^{(i)} - m = 0\}$;
return U

Let $Y = \left\{ y_j^{(i)} \mid i \in [r] \text{ and } j \in [s_i] \right\}$ be a new set of variables disjoint from X. For $i \in [r]$, let $g_1^{(i)}, \ldots, g_{s_i}^{(i)}$ be a topologically sorted enumeration of all nodes in circuit Φ_i. For all $j \in [s_i]$, let node $g_j^{(i)}$ be labelled by the variable $y_j^{(i)}$. Corresponding to each node in Φ_i, we shall now define a polynomial equation of degree at most 2 over the variable sets X and Y.

Algorithm 3 describes the construction of the extended system of equations. For each input node $g_j^{(i)}$, labeled by u (where u is either a variable x_k or a constant c) in Φ_i, the algorithm adds the polynomial equation $y_j^{(i)} - u = 0$ to the system T. For each product node $g_j^{(i)}$ with children labelled u and v (where u, v could be of the form $y_{j'}^{(i)}$ for some $j > j'$ or $x_{k'}$), the algorithm introduces a new polynomial equation $y_j^{(i)} - u \cdot v = 0$ to the system T, and for each sum node $g_j^{(i)}$ with children labelled $u_1, \ldots, u_k$ (where $u_1, \ldots, u_k$ could be of the form $y_{j'}^{(i)}$ for some $j > j'$ or $x_{k'}$), it introduces a new polynomial equation $y_j^{(i)} - \sum_{i=1}^{k} u_i = 0$. At the termination of the algorithm, the system T consists of either constant-free quadratic binomial equations or affine linear polynomial equations. It is also clear that there exists $\mathbf{a}'$ that satisfies T if and only if there is $\mathbf{a}$ that satisfies S. As before this is easy to see: by following the flow of computation in an arithmetic circuit from leaves to the root, we can infer the values of $y_j^{(i)}$ for all $i \in [t]$ and $j \in [s_i]$. This discussion can be summarized as Lemma 20 below.

▶ **Lemma 20** (Lemma 8 restated). *Let $S = \{f_i(X) = 0\}_{i=1}^{r}$ be a system of polynomial equations over the polynomial ring $R[X]$ such that for each $i \in [r]$, the polynomial $f_i(X)$ is provided as an arithmetic circuit Φ_i of size s_i. Then, when given this as input, Algorithm 3 runs in time $\mathrm{poly}(|X|, r, \max_i\{s_i\})$ and returns a system T of polynomial equations $\{g_j(X, Y) = 0\}_{j=1}^{t}$ over the polynomial ring $R[X, Y]$ such that*

- $t = \mathrm{poly}(|X|, r, \max_i\{s_i\})$.
- $|Y| \leq r \cdot \max_i\{s_i\}$.
- *Each polynomial equation $g_j(X, Y) = 0$ in T is either a quadratic binomial polynomial equation or an affine linear polynomial equation.*
- *T has a solution in $R^{|X \sqcup Y|}$ if and only S has a solution in $R^{|X|}$.*

Algorithm 3 ConstructExtendedSystem-CircuitRepresentation(S).

Result: Given a system $S = \{f_1 = 0, \ldots, f_r = 0\}$ of polynomial equations provided in white-box circuit representation, we construct a system T of polynomial equations over an extended set of variables such that each polynomial in T has degree of at most 2 and if $\mathbf{a}$ solves S then there exists an extension $\mathbf{a}'$ of $\mathbf{a}$ such that $\mathbf{a}'$ solves T and vice-versa.

$T \leftarrow \emptyset$;
for $i \in [r]$ **do**
  Let Φ_i be the circuit computing f_i, and $s_i = |\Phi_i|$;
  for $j \in [s_i]$ **do**
    if $g_j^{(i)}$ *is a leaf node in* Φ_i **then**
      Let u be a variable or a constant labeling the input node in Φ_i;
      $T \leftarrow T \cup \{y_j^{(i)} - u = 0\}$;
    else
      if $g_j^{(i)}$ *is a product node in* Φ_i **then**
        Let u and v be the renaming of the labels of the children of $g_j^{(i)}$;
        $T \leftarrow T \cup \{y_j^{(i)} - u \cdot v = 0\}$;
      else
        $g_j^{(i)}$ is a sum node in Φ_i;
        Let $u_1, \ldots, u_k$ be the renaming of the labels of the children of $g_j^{(i)}$;
        $T \leftarrow T \cup \{y_j^{(i)} - \sum_{i=1}^{k} u_i = 0\}$;
      end
    end
  end
end
return T;