On the Symmetries of and Equivalence Test for Design Polynomials

Nikhil Gupta
Department of Computer Science and Automation, Indian Institute of Science, India
nikhilg@iisc.ac.in

Chandan Saha
Department of Computer Science and Automation, Indian Institute of Science, India
chandan@iisc.ac.in

Abstract

In a Nisan-Wigderson design polynomial (in short, a design polynomial), every pair of monomials share a few common variables. A useful example of such a polynomial, introduced in [34], is the following:

$$\mathsf{NW}_{d,k}(\mathbf{x}) = \sum_{h \in \mathbb{F}_d[z],\ \deg(h) \leq k} \ \prod_{i=0}^{d-1} x_{i,h(i)},$$

where d is a prime, $\mathbb{F}_d$ is the finite field with d elements, and $k \ll d$. The degree of the gcd of every pair of monomials in $\mathsf{NW}_{d,k}$ is at most k. For concreteness, we fix $k = \lceil \sqrt{d} \rceil$. The family of polynomials $\mathcal{NW} := \{\mathsf{NW}_{d,k} : d \text{ is a prime}\}$ and close variants of it have been used as hard explicit polynomial families in several recent arithmetic circuit lower bound proofs. But, unlike the permanent, very little is known about the various structural and algorithmic/complexity aspects of $\mathcal{NW}$ beyond the fact that $\mathcal{NW} \in \mathsf{VNP}$. Is $\mathsf{NW}_{d,k}$ characterized by its symmetries? Is it circuit-testable, i.e., given a circuit C can we check efficiently if C computes $\mathsf{NW}_{d,k}$? What is the complexity of equivalence test for $\mathcal{NW}$, i.e., given black-box access to a $f \in \mathbb{F}[\mathbf{x}]$, can we check efficiently if there exists an invertible linear transformation A such that $f = \mathsf{NW}_{d,k}(A \cdot \mathbf{x})$? Characterization of polynomials by their symmetries plays a central role in the geometric complexity theory program. Here, we answer the first two questions and partially answer the third.

We show that $\mathsf{NW}_{d,k}$ is characterized by its group of symmetries over $\mathbb{C}$, but not over $\mathbb{R}$. We also show that $\mathsf{NW}_{d,k}$ is characterized by circuit identities which implies that $\mathsf{NW}_{d,k}$ is circuit-testable in randomized polynomial time. As another application of this characterization, we obtain the "flip theorem" for $\mathcal{NW}$.

We give an efficient equivalence test for $\mathcal{NW}$ in the case where the transformation A is a block-diagonal permutation-scaling matrix. The design of this algorithm is facilitated by an almost complete understanding of the group of symmetries of $\mathsf{NW}_{d,k}$: We show that if A is in the group of symmetries of $\mathsf{NW}_{d,k}$ then $A = D \cdot P$, where D and P are diagonal and permutation matrices respectively. This is proved by completely characterizing the Lie algebra of $\mathsf{NW}_{d,k}$, and using an interplay between the Hessian of $\mathsf{NW}_{d,k}$ and the evaluation dimension.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Nisan-Wigderson design polynomial, characterization by symmetries, Lie algebra, group of symmetries, circuit testability, flip theorem, equivalence test

Digital Object Identifier 10.4230/LIPIcs.MFCS.2019.53

Acknowledgements We would like to thank Neeraj Kayal, Meena Mahajan for some insightful discussions on the design polynomial family. NG would also like to thank Anuj Tawari for his time in sitting through a few presentations on the proof of Theorem 4. We also thank anonymous reviewers for their comments.

© Nikhil Gupta and Chandan Saha;
licensed under Creative Commons License CC-BY
44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).
Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 53; pp. 53:1–53:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Proving super-polynomial lower bounds for Boolean and arithmetic circuits computing explicit functions is the holy grail of circuit complexity. Over the past few decades, research on lower bounds has gradually pushed the frontier by bringing in novel methods in the arena and carefully building upon the older ones. Some of the notable achievements are – lower bounds for AC^0 circuits [2, 17, 26], monotone circuits [3, 57], $\mathsf{ACC}(p)$ circuits [58, 62] and ACC circuits [52, 63] in the Boolean case, and lower bounds for homogeneous depth three circuits [54], multilinear formulas [55, 56], homogeneous depth four circuits [23, 31, 42] and the lower bound on the depth of circuits for MaxFlow [46] in the arithmetic case. The slow progress in circuit lower bounds is explained by a few "barrier" type results, particularly by the notion of natural proofs [59] for Boolean circuits, and the notion of algebraically natural proofs [13, 21] for arithmetic circuits [1]. Most lower bound proofs, but not all [2], do fit in the natural proof framework.

It is apparent from the concept of natural proofs and its algebraic version that in order to avoid this barrier, we need to develop an approach that violates the so called *constructivity* criterion or the *largeness* criterion. Focusing on the latter criterion, it means, if an explicit function has a special property that random functions do not have, and if a lower bound proof for circuits computing this explicit function uses this special property critically, then such a proof circumvents the natural proof barrier automatically. For polynomial functions (simply polynomials), *characterization by symmetries* is such a special property[3], and the geometric complexity theory (GCT) program [51] is an approach to proving super-polynomial arithmetic circuit lower bound by crucially exploiting this property of the permanent and the determinant polynomials. From hereon, our discussion will be restricted to polynomial functions and arithmetic circuits.

The permanent family is complete for the class VNP and the determinant family is complete for the class VBP under p-projections. The class $\mathsf{VBP} \subseteq \mathsf{VP}$ consists of polynomial families that are computable by poly-size algebraic branching programs; this class has another interesting complete family, namely the iterated matrix multiplication (IMM) family. These three polynomial families have appeared in quite a few lower bound proofs [9, 15, 20, 23, 36, 42, 45, 54–56] in the arithmetic circuit literature. That permanent and determinant are characterized by their respective groups of symmetries are classical results [16, 44]. It has also been shown that IMM is characterized by its symmetries [19, 32]. There are two other polynomial families in VP, the power symmetric polynomials and the sum-product polynomials, that are known to possess this rare property (see Section 2 in [8]). However, the elementary symmetric polynomial is not characterized by its symmetries [27].

In the recent years, another polynomial, namely the Nisan-Wigderson design polynomial (in short, design polynomial), and close variants of it have been used intensely as hard explicit polynomials in several lower bound proofs for depth three, depth four and depth five circuits [10, 12, 31, 33–35, 37–42]. In some cases, the design polynomial (Definition 7) yielded lower bounds that are not known yet for the permanent, determinant and IMM (as

[1] Presently, the evidences in favor of existence of one-way functions (which implies the natural proof barrier) are much stronger than that of existence of succinct hitting-set generators (which implies the algebraically natural proof barrier). However, there are a few results in algebraic complexity that exhibit, unconditionally [11] or based on more plausible complexity theoretic assumptions [5], the limitations of some of the current techniques in proving lower bounds for certain restricted arithmetic models.

[2] like the lower bounds for monotone and ACC circuits

[3] A random polynomial is not characterized by its symmetries with high probability (see Proposition 3.4.9 in [22])

in [12, 33, 37, 40]). It can be easily shown that the design polynomial defines a family in VNP (see Observation B.1 in [24]). But, very little is otherwise known about the various structural and algorithmic/complexity aspects of this family. Like the permanent, is it characterized by its symmetries? Is it circuit testable? What is the complexity of equivalence test for the Nisan-Wigderson design polynomial? It is reasonable to seek answers to these fundamental questions for a natural family like the design polynomials. Moreover, in the light of some recent developments in GCT [7, 28, 29], it may be worth studying other polynomial families (like the design polynomials and the IMM) that have some of the "nice features" of the permanent and the determinant and that may also fit in the GCT framework. We refer the reader to [1, 22, 50, 60] for an overview of GCT. If the design polynomial family turns out to be in VP then that would be an interesting result by itself with potentially important complexity theoretic and algorithmic consequences. If a polynomial has a small depth-4 circuit, then it is a projection of a small NW design polynomial (see Observation B.2 in [24])

In this article, we answer some of the above questions on the design polynomial pertaining to its group of symmetries. Our results accord a fundamental status to this polynomial.

1.1 Our results

Some of the basic definitions and notations are given in Section 2. The design polynomial $\mathsf{NW}_{d,k}$ is defined (in Definition 7) using two parameters, d (the degree) and k (the "intersection" parameter). Our results hold for any $k \in [1, \frac{d}{4} - 5]$, but (from the lower bound point of view) it is best to think of k as d^{ϵ} for some arbitrarily chosen constant $\epsilon \in (0, 1)$. The number of variables in $\mathsf{NW}_{d,k}$ is $n = d^2$. Any polynomial can be expressed as an affine projection of $\mathsf{NW}_{d,k}$, for a possibly large d (see Observation B.2 in [24]). For notational convenience, we will drop the subscripts d and k whenever they are clear from the context. Let $\mathscr{G}_f$ be the group of symmetries of a polynomial f over an underlying field $\mathbb{F}$ (see Definition 12).

▶ **Theorem 1** (Characterization by symmetries). *Let $\mathbb{F} = \mathbb{C}$ and f be a homogeneous degree-d polynomial in $n = d^2$ variables. If $\mathscr{G}_{\mathsf{NW}} \subseteq \mathscr{G}_f$ then $f = \alpha \cdot \mathsf{NW}$ for some $\alpha \in \mathbb{C}$.*

The theorem, proven in Section 3, holds over any field $\mathbb{F}$ having a d-th root of unity $\zeta \neq 1$ and $|\mathbb{F}| \neq d + 1$. We also show in Section 4.3 that NW is not characterized by its symmetries over $\mathbb{R}, \mathbb{Q}$ and finite fields not containing a d-th primitive root of unity – in contrast, the permanent is characterized by its symmetries over these fields. The symmetries of NW have a nice algorithmic application: Although, it is not known if NW is computable by a poly(d) size circuit (Definition 6), the following theorem shows that checking if a given circuit computes NW can be done efficiently. In this article, whenever we mention size-s circuit, we mean size-s circuit with degree bounded by $\delta(s)$, which is an arbitrarily fixed polynomial function[4] of s. Let $\mathbf{x}$ be the set of n variables of NW. We will identify a circuit with the polynomial computed by it.

▶ **Theorem 2** (Circuit testability). *There is a randomized algorithm that takes input as black-box access to a circuit $\mathsf{C}(\mathbf{x})$ of size s over a finite field $\mathbb{F}$, where $|\mathbb{F}| \geq 4 \cdot \delta(s)$ (recall $\delta(s)$ is an upper bound on the degree of size s circuits), and determines correctly whether or not $\mathsf{C}(\mathbf{x}) = \mathsf{NW}$ with high probability, using* poly(s) *field operations.*

[4] This is the interesting scenario in algebraic complexity theory as polynomial families in VP admit circuits with degree bounded by a polynomial function of size.

A suitable version of the theorem also holds over $\mathbb{Q}, \mathbb{R}$ and $\mathbb{C}$. Such a theorem is known for the permanent with two different proofs, one using self-reducibility of the permanent [43] and the other using its symmetries [48]. We do not know if NW has a self-reducible property like the permanent, but its symmetries are powerful enough to imply the above result. The theorem is proven in Section 5 by showing that NW is characterized by circuit identities over *any* field (see Definition 18). This characterization, which uses the symmetries of NW, also implies the following result. For this result, we can assume $\delta(s) \geq d$, without any loss of generality.

▶ **Theorem 3** (Flip theorem). *Suppose* NW *is not computable by circuits of size s over a finite field $\mathbb{F}$, where $|\mathbb{F}| \geq 4 \cdot \delta(s)$ and $\delta(s)$ is an upper bound on the degree of size s circuits. Then, there exist points $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{F}^n$, where $m = \mathrm{poly}(s)$, such that for every circuit C over $\mathbb{F}$ of size at most s, there is an $\ell \in [m]$ satisfying $\mathsf{C}(\mathbf{a}_\ell) \neq \mathsf{NW}(\mathbf{a}_\ell)$. A set of randomly generated points $\mathbf{a}_1, \ldots, \mathbf{a}_m \in_r \mathbb{F}^n$ has this property with high probability. Moreover, black-box derandomization of polynomial identity testing for size-$(10s)$ circuits over $\mathbb{F}$ using $\mathrm{poly}(s)$ field operations implies that the above-mentioned points can be computed deterministically using $\mathrm{poly}(s)$ field operations.*

An appropriate version of the theorem also holds over $\mathbb{Q}, \mathbb{R}$ and $\mathbb{C}$. The flip theorem is known for the permanent [48, 49] [5]. Similar theorems have also been shown for the $3SAT$ problem [4, 14]. Results of this kind show that if a certain function ($3SAT$ or permanent or NW) is not computable by small circuits then there exists a short list of efficiently computable “hard instances” that fail all small circuits.
We show another algorithmic application of the knowledge of the symmetries of NW in solving a natural case of the equivalence test problem for NW, namely block-diagonal permutation-scaling equivalence test (BD-PS equivalence test, in short). An equivalence test for NW checks if a given polynomial $f \in \mathbb{F}[\mathbf{x}]$ satisfies $f = \mathsf{NW}(A \cdot \mathbf{x})$, where A is an invertible linear transformation. A BD-PS equivalence test is the special case where A is a product of a block-diagonal permutation matrix and an invertible scaling matrix. The following theorem is proved in Section 6.

▶ **Theorem 4** (BD-PS equivalence test for NW). *Let $k \in [1, \frac{d}{3}]$, $\mathbb{F}$ be a finite field such that $d \nmid (|\mathbb{F}| - 1)$ and $|\mathbb{F}| \geq 4d$. There is a randomized algorithm that takes input black-box access to a degree d polynomial $f \in \mathbb{F}[\mathbf{x}]$ and correctly decides if f is BD-PS equivalent to* NW *with high probability. If the answer is yes then it outputs a A such that $f = \mathsf{NW}(A \cdot \mathbf{x})$, where A is a product of a block-diagonal permutation matrix and an invertible scaling matrix. The running time is* $\mathrm{poly}(d, \log |\mathbb{F}|)$.

An appropriate version of the theorem holds over $\mathbb{R}$ (details given in Section F.4 of [24]). Efficient equivalence tests are known for the Permanent and IMM over $\mathbb{C}$, $\mathbb{Q}$ and finite fields [30, 32] and for the Determinant over $\mathbb{C}$ and finite fields [18, 30]. In [30], it was shown that equivalence test for the Permanent reduces to permutation-scaling (PS) equivalence test. We show in Section 6 that equivalence test for NW reduces to block-permuted equivalence test[6],i.e., we can assume without loss of generality that A is a block-permuted matrix. Theorem 4 solves the equivalence test for NW in the case where A is a block-diagonal matrix and additionally has the permutation-scaling (PS) structure. Even this case is quite nontrivial and may serve as an important ingredient for an efficient general equivalence test for NW.

[5] We have borrowed the name “flip theorem” from these work.
[6] It decides if there exists a block-permuted matrix (Definition 8) $A \in \mathrm{GL}_{d^2}(\mathbb{F})$ such that $f = \mathsf{NW}(A \cdot \mathbf{x})$

The design of the test in Theorem 4 is facilitated by a near complete understanding of the symmetries of NW as stated in the following theorem. The proof is given in Section 4.2.

▶ **Theorem 5** (Structure of $\mathscr{G}_{\mathsf{NW}}$). *Let $\mathbb{F}$ be the underlying field of size greater than $\binom{d}{2}$ and* $\mathrm{char}(\mathbb{F}) \neq d$. *If $A \in \mathscr{G}_{\mathsf{NW}}$ then $A = D \cdot P$, where $D, P \in \mathscr{G}_{\mathsf{NW}}$ are diagonal and permutation matrices respectively.*

The group of symmetries of the permanent has a similar structure [44]. The above structure also plays a crucial role in showing that NW is not characterized by its symmetries over $\mathbb{R}$. The proof of the theorem involves a complete characterization of the Lie algebra of NW, and an interplay between the Hessian of NW and the evaluation dimension measure. We first prove the structural results (Theorems 1 and 5) and then show their algorithmic applications (Theorems 2, 3 and 4). The proof details are shifted to the appendix. A comparison between the Permanent and NW is summarized in a table in Section A of [24].

2 Preliminaries

Notations. The set of natural numbers is $\mathbb{N} = \{0, 1, 2 \ldots\}$ and $\mathbb{N}^\times = \mathbb{N} \backslash \{0\}$. For $r \in \mathbb{N}^\times$, $[r] = \{0, \ldots, r-1\}$. The general linear group $\mathrm{GL}_r(\mathbb{F})$ is the group of all $r \times r$ invertible matrices over $\mathbb{F}$. Throughout this article, $\mathrm{poly}(r)$ means $r^{O(1)}$ and $\exp(r)$ means 2^r. For a prime d, $\mathbb{F}_d$ is the finite field of order d whose elements are naturally identified with $[d] = \{0, 1, \ldots, d-1\}$. Let $\mathbf{x}$ be the following disjoint union of variables,

$$\mathbf{x} := \biguplus_{i \in [d]} \mathbf{x}_i, \tag{1}$$

where $\mathbf{x}_i := \{x_{i,0}, \ldots, x_{i,d-1}\}$. The total number of variables in $\mathbf{x}$ is $n = d^2$. $\mathbb{F}[\mathbf{x}]$ and $\mathbb{F}_d[z]$ denote the rings of multivariate and univariate polynomials over $\mathbb{F}$ and $\mathbb{F}_d$ in $\mathbf{x}$ and z variables respectively, and the set $\mathbb{F}_d[z]_k := \{h \in \mathbb{F}_d[z] : \deg(h) \leq k\}$. We will represent elements of $\mathbb{F}$ by lower case Greek alphabets $(\alpha, \beta, ...)$, elements of $\mathbb{F}_d$ by lower case Roman alphabets $(a, b, ...)$, multivariate polynomials over $\mathbb{F}$ by f, g and q, univariate polynomials over $\mathbb{F}_d$ by p and h, matrices over $\mathbb{F}$ by capital letters $(A, B, C, ...)$, and the set of variables by $\mathbf{x}, \mathbf{y}, \mathbf{z}$ and vectors over $\mathbb{F}$ by $\mathbf{a}, \mathbf{b}$. Variable sets are interpreted as column vectors when left multiplied to a matrix. For instance, in $A \cdot \mathbf{x}$, $\mathbf{x}$ is the vector $(x_{0,0}\ x_{0,1} \ldots\ x_{0,d-1}\ \ldots\ x_{d-1,0}\ x_{d-1,1} \ldots\ x_{d-1,d-1})^T$, and we say A is applied on $\mathbf{x}$.

2.1 Algebraic preliminaries

A polynomial f is homogeneous if the degree of all the monomials of f are the same. Polynomial $f \in \mathbb{F}[\mathbf{x}]$ is set-multilinear in the sets $\mathbf{x}_0, \ldots, \mathbf{x}_{d-1}$ (as defined in Equation (1)) if every monomial contains exactly one variable from each set $\mathbf{x}_i$ for $i \in [d]$.

▶ **Definition 6** (Arithmetic circuit). *An arithmetic circuit* C *over $\mathbb{F}$ is a directed acyclic graph in which a node with in-degree zero is labelled with either a variable or an $\mathbb{F}$-element, an edge is labelled with an $\mathbb{F}$-element, and other nodes are labelled with $+$ and $\times$. Computation proceeds in a natural way: a node with in-degree zero computes its label, an edge scales a polynomial by its label, and a node labelled with $+/\times$ computes the sum/product of the polynomials computed at the end of the edges entering the node. The polynomials computed by nodes with out-degree zero are the outputs of* C. *The size of* C *is the sum of the number of nodes and edges in the graph. The* degree *of* C *is the maximum over the degree of the polynomials computed at all nodes of* C.

▶ **Definition 7** (Nisan-Wigderson polynomial). *Let $d > 2$ be a prime and $k \in \mathbb{N}$. The Nisan-Wigderson design polynomial is defined as in [34] (which is inspired by the Nisan-Wigderson set-systems [53]),*

$$\mathsf{NW}_{d,k}(\mathbf{x}) := \sum_{h \in \mathbb{F}_d[z]_k} \prod_{i \in \mathbb{F}_d} x_{i,h(i)}.$$

It is a degree-d homogeneous and set-multilinear polynomial in $n = d^2$ variables, having d^{k+1} monomials. We drop the subscripts d, k for notational convenience. NW satisfies the "low intersection" property, meaning any two monomials of NW have at most k variables in common. This follows because the monomials are obtained from polynomials in $\mathbb{F}_d[z]_k$.

▶ **Definition 8** (Block-permuted matrix). *A matrix $A \in \mathbb{F}^{d^2 \times d^2}$ is a block-permuted matrix with block size d if $A = B \cdot (P \otimes I_d)$, where $B \in \mathbb{F}^{d^2 \times d^2}$ is a block-diagonal matrix with block size d, $P \in \mathbb{F}^{d \times d}$ is a permutation matrix, and I_d is the $d \times d$ identity matrix.*

▶ **Definition 9** (Evaluation dimension). *Let $f \in \mathbb{F}[\mathbf{y}]$ and $\mathbf{z} \subseteq \mathbf{y}$. The evaluation dimension of f with respect to $\mathbf{z}$ is, $evalDim_{\mathbf{z}}(f) := \dim(\ \mathbb{F}\text{-}span\ \{f(\mathbf{y})|_{\mathbf{z}=\mathbf{a}} : \boldsymbol{a} \in \mathbb{F}^{|\mathbf{z}|}\}\ )$.*

▶ **Definition 10** (Hessian). *Let $f \in \mathbb{F}[\mathbf{y}]$ be a polynomial in $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ variables. The Hessian of f is the following matrix in $(\mathbb{F}[\mathbf{y}])^{n \times n}$,*

$$H_f(\mathbf{y}) := \left(\frac{\partial^2 f}{\partial y_i \cdot \partial y_j}\right)_{i,j \in [n]}.$$

The following property of $H_f(\mathbf{y})$ that can be proved using chain-rule of derivatives.

▶ **Lemma 11** (Lemma 2.6 of [8]). *Let $g \in \mathbb{F}[\mathbf{y}]$ and $f = g(A \cdot \mathbf{y})$ for some $A \in \mathbb{F}^{n \times n}$. Then, $H_f(\mathbf{y}) = A^T \cdot H_g(A \cdot \mathbf{y}) \cdot A$.*

▶ **Definition 12** (Group of symmetries). *Let $f \in \mathbb{F}[\mathbf{y}]$ be an n-variate polynomial. The set $\mathscr{G}_f = \{A \in \mathrm{GL}_n(\mathbb{F}) : f(A \cdot \mathbf{y}) = f(\mathbf{y})\}$ forms a group under matrix multiplication and it is called the group of symmetries of f over $\mathbb{F}$.*

▶ **Definition 13** (Lie algebra). *Let $f \in \mathbb{F}[\mathbf{y}]$ be a polynomial in $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ variables. The Lie algebra of f, denoted by $\mathfrak{g}_f$, is the set of matrices $B = (b_{i,j})_{i,j \in [n]} \in \mathbb{F}^{n \times n}$ satisfying the relation $\sum_{i,j \in [n]} b_{i,j} \cdot y_j \cdot \frac{\partial f}{\partial y_i} = 0$.*

It is easy to check that $\mathfrak{g}_f$ is a vector space over $\mathbb{F}$. The following property relates the Lie algebras of $f(\mathbf{y})$ and $f(A \cdot \mathbf{y})$ for $A \in \mathrm{GL}_n(\mathbb{F})$. See Proposition 58 of [30] for its proof.

▶ **Lemma 14** (Conjugacy of Lie algebras). *Let $g \in \mathbb{F}[\mathbf{y}]$ be an n-variate polynomial. If $f(\mathbf{y}) = g(A \cdot \mathbf{y})$ for $A \in \mathrm{GL}_n(\mathbb{F})$, then $\mathfrak{g}_f = A^{-1} \cdot \mathfrak{g}_g \cdot A$.*

▶ **Lemma 15.** *[30] Given black-box access to an n-variate degree d polynomial $f \in \mathbb{F}[\mathbf{x}]$, a basis of $\mathfrak{g}_f$ can be computed in randomized $poly(n, d, \rho)$ time, where ρ is the bit complexity of the coefficients of f.*

Over $\mathbb{C}$, the Lie algebra $\mathfrak{g}_f$ is related to the group of symmetries $\mathscr{G}_f$ as stated in the following definition. For $B \in \mathbb{C}^{n \times n}$, let $e^B := \sum_{i \in \mathbb{N}} \frac{B^i}{i!} \in \mathbb{C}^{n \times n}$ (the series always converges).

▶ **Definition 16** (Continuous and discrete symmetries). *Let $f \in \mathbb{C}[\mathbf{y}]$. If $A \in \mathfrak{g}_f$ then $e^{tA} \in \mathscr{G}_f$ for every $t \in \mathbb{R}$ (see [25] for a proof of this fact). Elements of the set $\{e^{tA} : A \in \mathfrak{g}_f \text{ and } t \in \mathbb{R}\}$ are the continuous symmetries of f. All the other symmetries in $\mathscr{G}_f$ are the discrete symmetries of f.*

▶ **Definition 17** (Characterization by symmetries). *A homogeneous degree-d polynomial $g \in \mathbb{F}[\mathbf{y}]$ is said to be characterized by its symmetries if for every degree-d homogeneous polynomial $f \in \mathbb{F}[\mathbf{y}]$, $\mathscr{G}_g \subseteq \mathscr{G}_f$ implies that $f(\mathbf{y}) = \alpha \cdot g(\mathbf{y})$ for some $\alpha \in \mathbb{F}$.*

▶ **Definition 18** (Characterization by circuit identities). *Let $g \in \mathbb{F}[\mathbf{y}]$ be an n-variate polynomial, and $\mathbf{z}, \mathbf{u}$ be two sets of constantly many variables and $|\mathbf{z}| = c$. Suppose that there exist $m = \text{poly}(n)$ polynomials $q_1(\mathbf{z}, \mathbf{u}), \ldots, q_m(\mathbf{z}, \mathbf{u})$ over $\mathbb{F}$ such that for every $i \in [m]$, q_i is computable by a constant size circuit and there exist $A_{i1}, \ldots, A_{ic} \in \mathbb{F}[\mathbf{u}]^{n \times n}$ computable by $\text{poly}(n)$ size circuits, and the following condition is satisfied: For $f \in \mathbb{F}[\mathbf{y}]$, $q_i(f(A_{i1} \cdot \mathbf{y}), \ldots, f(A_{ic} \cdot \mathbf{y}), \mathbf{u}) = 0$ for every $i \in [m]$ if and only if $f = \alpha \cdot g$ for some $\alpha \in \mathbb{F}$. Then, g is* characterized by circuit identities *over $\mathbb{F}$.*

The above definition is taken (after slight modifications to suit our purpose) from Definition 3.4.7 in [22] and is attributed to an article by Mulmuley [47].

3 Characterization of NW by symmetries and circuit identities

3.1 Symmetry characterization: Theorem 1

Let $\mathbb{F}$ be a field having a d-th root of unity $\zeta \neq 1$ and $|\mathbb{F}| \neq d+1$.[7] As d is a prime, ζ is primitive, i.e., $\zeta^d = 1$ and $\zeta^t \neq 1$ for $0 < t < d$. The rows and columns of a matrix in $\mathscr{G}_{\mathsf{NW}}$ are indexed by the set $\{(i,j) : i, j \in \mathbb{F}_d\}$.

▷ Claim 19. The following matrices in $\mathbb{F}^{n \times n}$ are in $\mathscr{G}_{\mathsf{NW}}$:

1. $A_{\boldsymbol{\beta}}$, a diagonal matrix with $A_{\beta}((i,j),(i,j)) = \beta_i \in \mathbb{F}^{\times}$ for $i, j \in [d]$, s.t. $\prod_{i \in [d]} \beta_i = 1$.
2. A_ℓ, a diagonal matrix with $((i,j),(i,j))$-th entry as $\zeta^{i^\ell \cdot j}$ for $i, j \in [d]$ and $\ell \in [d-k-1]$.[8]
3. A_h, $h \in \mathbb{F}_d[z]_k$,such that $A_h((i,j),(i,j+h(i))) = 1$ for $i, j \in [d]$ and other entries are 0.

The proof of Claim 19 is given in Section C.1 in [24]. The matrices $A_{\boldsymbol{\beta}}$ are the continuous symmetries while A_ℓ, A_h are discrete symmetries of NW for all choices of $\boldsymbol{\beta}, \ell, h$. The symmetries in 2 are very different from the symmetries of the Determinant and the Permanent. The following Claim immediately implies Theorem 1. Its proof is given in Section C.2 in [24].

▷ Claim 20. Let f be a homogeneous degree-d polynomial in $\mathbb{F}[\mathbf{x}]$. If $\mathscr{G}_f$ contains $A_{\boldsymbol{\beta}}, A_\ell$ and A_h (for all choices of $\boldsymbol{\beta}$, ℓ and h, mentioned above) then $f = \alpha \cdot \mathsf{NW}$ for some $\alpha \in \mathbb{F}$.

3.2 Characterization by circuit identities

Here we show that NW is characterized by circuit identities (Definition 18). The lemma is crucially used to prove Theorems 2 and 3 in Section 5. Its proof is given in Section C.3 in [24].

▶ **Lemma 21.** *Polynomial* NW *is characterized by circuit identities over any field $\mathbb{F}$.*

4 Lie algebra and symmetries of NW

We first give a complete description of the Lie algebra of NW by giving an explicit $\mathbb{F}$-basis. Then, using this knowledge, we analyse the structure of the symmetries of NW and prove

[7] For a prime d, $|\mathbb{F}| = d+1$ if and only if d is a Mersenne prime.
[8] Recall, $[d-k-1] = \{0, 1, \ldots, d-k-2\}$

Theorem 5. Thereafter, using Theorem 5, we show that NW is not characterized by its symmetries over fields that do not contain a d-th primitive root of unity. The rows and columns of a $n \times n$ matrix in $\mathfrak{g}_{\mathsf{NW}}$ and $\mathscr{G}_{\mathsf{NW}}$ are indexed by the set $\{(i,j) : i,j \in \mathbb{F}_d\}$, which is naturally identified with the $\mathbf{x}$-variables, where $\mathbf{x} = (x_{0,0} \dots x_{0,d-1} \ \dots \ x_{d-1,0} \dots x_{d-1,d-1})^T$.

4.1 Lie algebra of NW

It turns out that the Lie algebra of NW is a subspace of the Lie algebra of every set-multilinear polynomial. (The default partition of a set-multilinear polynomial is $\mathbf{x} = \uplus_{i\in[d]}\mathbf{x}_i$.)

▶ **Lemma 22.** *Let $\mathbb{F}$ be a field and $\mathrm{char}(\mathbb{F}) \neq d$. The dimension of $\mathfrak{g}_{\mathsf{NW}}$ over $\mathbb{F}$ is $d-1$, and the diagonal matrices $B_1, \dots, B_\ell$ (defined below) form a $\mathbb{F}$-basis of $\mathfrak{g}_{\mathsf{NW}}$. For $\ell \in \{1, \dots, d-1\}$,*

$$(B_\ell)_{(i,j),(i,j)} = \begin{cases} 1, & \text{if } i = 0, j \in [d] \\ -1, & \text{if } i = \ell, j \in [d] \\ 0, & \text{otherwise.} \end{cases}$$

The lemma is proven in Section D.1 in [24] by carefully analysing a system of linear equations obtained from the monomials of NW. It follows that every $B \in \mathfrak{g}_{\mathsf{NW}}$ is of the form $\mathrm{diag}(\alpha_0, \dots, \alpha_{d-1}) \otimes I_d$, where each $\alpha_i \in \mathbb{F}$ and $\sum_{i\in[d]} \alpha_i = 0$. The continuous symmetries of NW consist of matrices $A = \mathrm{diag}(\beta_0, \dots, \beta_{d-1}) \otimes I_d$, where each $\beta_i \in \mathbb{C}$ and $\prod_{i\in[d]} \beta_i = 1$.

4.2 Structure of $\mathscr{G}_{\mathsf{NW}}$: Theorem 5

Lemma 22 implies the following.

▷ Claim 23. Every $A \in \mathscr{G}_{\mathsf{NW}}$ is a block-permuted matrix with block size d.

The proof of the claim is given in Section D.2 in [24]. Using Claim 23, Hessian and the evaluation dimension of NW, we give a proof of Theorem 5 in Section D.3 in [24].

4.3 NW is not characterized by its symmetries over $\mathbb{R}$

Let $\mathbb{F}$ be either $\mathbb{R}, \mathbb{Q}$ or a finite field such that $d \nmid |\mathbb{F}| - 1$. Then, $\mathbb{F}$ does not contain a d-th primitive root of unity, and so the matrices A_ℓ, for $\ell \in [d-k-1]$ mentioned in Claim 19, are no longer the symmetries of NW over $\mathbb{F}$. The next lemma shows that over such $\mathbb{F}$ all the diagonal symmetries of NW are of the type $A_{\boldsymbol{\beta}}$ mentioned in Claim 19. This then implies the following theorem, which may seem somewhat surprising as we do not know all the permutation symmetries of NW. The proofs are given in Section D.4 in [24].

▶ **Lemma 24.** *If $D \in \mathscr{G}_{\mathsf{NW}}$ is a diagonal matrix over $\mathbb{F}$ then $D = diag(\beta_0, \dots, \beta_{d-1}) \otimes I_d$, where each $\beta_i \in \mathbb{F}$ and $\prod_{i\in[d]} \beta_i = 1$.*

▶ **Theorem 25.** NW *is not characterized by its symmetries over $\mathbb{F}$.*

5 Circuit testability and the flip theorem for NW

In this and the next section, we show that the knowledge of the symmetries of NW plays a crucial role in answering some of the algorithmic questions related to NW. This section is devoted to Theorems 2 and 3. The main ingredient of their proofs is Lemma 21. We present the circuit testing algorithm here and push the proof of the Flip theorem to Section E in [24].

Proof of Theorem 2. Let $\mathtt{C}$ be a given circuit of size s over $\mathbb{F}$ that computes an n-variate polynomial $f = \mathtt{C}(\mathbf{x})$. Naturally, $\deg(f) \leq \delta(s)$. Algorithm 1 intends to check, in steps 2 and 3, if f satisfies the identities given in the proof of Lemma 21. If $f \neq \alpha \cdot \mathsf{NW}$ for all $\alpha \in \mathbb{F}$, then at least one of the identities is not satisfied. For the polynomials q_1, q_2 and q_3 defined in the proof of Lemma 21, observe that the degree of $q_1(f(A_i(u) \cdot \mathbf{x}), f(\mathbf{x}), u)$ is bounded by $2 \cdot \delta(s)$, whereas the degrees of $q_2(f(A_{a,r} \cdot \mathbf{x}), f(\mathbf{x}))$ and $q_3(f(A_t \cdot \mathbf{x}))$ are at most $\delta(s)$. As $|\mathbb{F}| \geq 4 \cdot \delta(s)$, by Schwartz-Zippel lemma [61, 64], step 4 returns "False" with probability at least $\frac{1}{2}$. If $f = \alpha \cdot \mathsf{NW}$ for some $\alpha \in \mathbb{F}$ then all the identities are satisfied, and step 7 ensures that $\alpha = 1$. Clearly, the algorithm uses $\text{poly}(s)$ field operations. The success probability is boosted from $\frac{1}{2}$ to $1 - \exp(-s)$ by repeating the algorithm $\text{poly}(s)$ times. ◀

Algorithm 1 Circuit testing for NW.

Input: Black-box access to a circuit $\mathtt{C}$ of size s over $\mathbb{F}$.
Output: "True" if $\mathtt{C}(\mathbf{x}) = \mathsf{NW}$, else "False".

1. Pick $\mathbf{a} \in_r \mathbb{F}^n$ and $\mu \in_r \mathbb{F}$.
2. **for** $i \in [d], a \in \mathbb{F}_d^{\times}, r \in [k+1], t \in [d] \backslash [k+1]$ **do**
3. **if** $(\mathtt{C}(A_i(\mu) \cdot \mathbf{a}) - \mu \cdot \mathtt{C}(\mathbf{a}) \neq 0)$ or $(\mathtt{C}(A_{a,r} \cdot \mathbf{a}) - \mathtt{C}(\mathbf{a}) \neq 0)$ or $(\mathtt{C}(A_t \cdot \mathbf{a}) \neq 0)$ **then**
4. **return** "False".
5. **end if**
6. **end for**
7. Let $\mathbf{b} \in \mathbb{F}^n$ be an assignment obtained by setting $x_{i0} = 1$, for $i \in [d]$, and all other variables to zero. If $f(\mathbf{b}) \neq 1$, return "False". Else, return "True".

6 Equivalence test for NW

First, we show a randomized reduction of equivalence test for NW to block-permuted equivalence test (in short, BP equivalence test) in Lemma 26. Then, we give an efficient equivalence test for NW in the special case where the linear transformation is block-diagonal and is a product of a permutation matrix and a scaling matrix (Theorem 4).

▶ **Lemma 26** (Reduction to BP equivalence test). *Let $\mathbb{F}$ be a field such that $\text{char}(\mathbb{F}) \neq d$ and $|\mathbb{F}| \geq 2d^2$. There is a randomized algorithm that takes input as black-box access to a degree d polynomial $f \in \mathbb{F}[\mathbf{x}]$ and does the following with high probability: It outputs black-box access to a degree d polynomial $g \in \mathbb{F}[\mathbf{x}]$ such that f is equivalent to NW if and only if g is BP equivalent to NW. Moreover, the transformation for f can be recovered efficiently from the transformation for g. The running time of this reduction is $\text{poly}(d, \rho)$, where ρ is bit complexity of the coefficients of f* [9].

Proof of correctness. The efficiency of Step 1 follows from Lemma 15. The correctness of Step 2 and 3 follow from the next claim whose proof is given in Section F.1 in [24].

▷ Claim 27. With high probability, matrix D can be computed in $\text{poly}(d, \rho)$ time. Moreover, f is equivalent to NW if and only if $f(D \cdot \mathbf{x})$ is BP equivalent to NW.

[9] We assume that univariate polynomial factorization over $\mathbb{F}$ can be done in polynomial time.

Algorithm 2 Reduction of equivalence test for NW to BP equivalence test.

Input: Black-box access to $f \in \mathbb{F}[\mathbf{x}]$.
Output: Black-box access to $g \in \mathbb{F}[\mathbf{x}]$.

1. Compute a basis $L_1, \ldots, L_r$ of $\mathfrak{g}_f$. If $r \neq d-1$, output "f is not equivalent to NW".
2. Let S be an arbitrary subset of $\mathbb{F}$ of size d^2. Let $L = a_1 L_1 + \ldots + a_r L_r$, where $a_i \in_r S$. Compute $D \in \mathrm{GL}_{d^2}(\mathbb{F})$ such that $D^{-1} \cdot L \cdot D = \mathrm{diag}(\beta_1, \ldots, \beta_d) \otimes I_d$, where $\beta_j \in \mathbb{F}$. If no such D exists then output "f is not equivalent to NW."
3. Output black-box access to $f(D \cdot \mathbf{x})$.

6.1 BD-PS equivalence test for NW: Theorem 4

Lemma 26 implies that to solve equivalence test for NW it is sufficient to focus on BP equivalence test. Here, we solve a special case of BP equivalence test, namely BD-PS equivalence test. We prove Theorem 4 in two steps: first we reduce BD-PS equivalence test to scaling equivalence test and then solve the scaling equivalence test. The algorithm pretends that f is BD-PS equivalent to NW and computes a block-diagonal permutation matrix A and an invertible scaling matrix B. In the end, the circuit testing algorithm of NW (Algorithm 1) is used to check if $f(A^{-1} \cdot B^{-1} \cdot \mathbf{x}) = \mathsf{NW}$.

6.1.1 Reduction of BD-PS equivalence test to scaling equivalence test

Assume $f = \mathsf{NW}(B \cdot A \cdot \mathbf{x})$, where A is a block-diagonal permutation matrix and B is an invertible scaling matrix. Algorithm 3 does not explicitly use the knowledge of the entries of B. Thus, we may assume without loss of generality that $B = I_{d^2}$. Then, the task reduces to solving the BD permutation equivalence test for NW. We identify matrix A with d permutations $\sigma_0, \ldots, \sigma_{d-1}$ on $[d]$ as $A = \mathrm{diag}(M_{\sigma_0}, \ldots, M_{\sigma_{d-1}})$, where M_{σ_i} is the $d \times d$ permutation matrix corresponding to σ_i [10].

▶ **Observation 6.1.** *Suppose f is BD permutation equivalent to* NW*, i.e. $f = \mathsf{NW}(A \cdot \mathbf{x})$. Then, a monomial $\prod_{i \in \mathbb{F}_d} x_{i,h(i)}$ of* NW *gets mapped to a unique monomial $\prod_{i \in \mathbb{F}_d} x_{i,\sigma_i(h(i))}$ of f.*

Algorithm 3 starts by assuming that $\sigma_0(0) = \cdots = \sigma_k(0) = 0$ and $\sigma_0(1) = 1$. The symmetries of NW allow us to make this assumption (Claim 28). The aim is to figure out all the entries of σ_i [11]. This is done by carefully picking a bunch of polynomials from $\mathbb{F}_d[z]_k$ (which we call *nice* polynomials) and then exploiting the association between f and NW mentioned in Observation 6.1 using these polynomials. The algorithm works over every field.

Proof of correctness. The following claims argue the correctness of the algorithm. Their proofs are given in Section F.2 in [24]. In these claims, ρ is the bit complexity of the coefficients of f.

▷ Claim 28. (Canonical form of $\sigma_0, \ldots, \sigma_{d-1}$): Suppose $f \in \mathbb{F}[\mathbf{x}]$ is BD permutation equivalent to NW. Then, there exist permutations $\sigma_0, \ldots, \sigma_{d-1}$ on $[d]$ such that $\sigma_0(0) = \cdots = \sigma_k(0) = 0, \sigma_0(1) = 1$ and $A = \mathrm{diag}(M_{\sigma_0}, \ldots, M_{\sigma_{d-1}})$ satisfies $f = \mathsf{NW}(A \cdot \mathbf{x})$.

[10] For $i, r, s \in [d], M_{\sigma_i}(r, s) = 1$ if and only if $\sigma_i(r) = s$.
[11] σ_i is treated as an ordered tuple $(\sigma_i(0), \ldots, \sigma_i(d-1))$

Algorithm 3 Block-diagonal permutation equivalence test for NW.

Input: Black-box access to $f \in \mathbb{F}[\mathbf{x}]$.
Output: Black-box access to $g \in \mathbb{F}[\mathbf{x}]$ such that if f is BD-PS equivalent to NW then g is scaling equivalent to NW.

1. Assume that $\sigma_0(0) = \cdots = \sigma_k(0) = 0$ and $\sigma_0(1) = 1$ (Claim 28).
2. Construct a list of nice polynomials in $\mathbb{F}_d[z]_k$ (Definition 29) as mentioned in Claim 30.
3. Recover $(d-k)$ distinct entries of each $\sigma_0, \ldots, \sigma_{d-1}$ as mentioned in Claim 31.
4. Let N be a $d \times d$ matrix, where the columns and rows are indexed by $(\sigma_0, \ldots, \sigma_{d-1})$ and $(0, \ldots, d-1)$ respectively and for $l, i \in [d], N(l, i) := \sigma_i(l)$. Pick $l_0, \ldots, l_k \in [d]$ such that in each of the rows indexed by $l_0, \ldots, l_k$ at least $k+1$ entries are known (Claim 32).
5. Use $l_0, \ldots, l_k \in [d]$ to recover all the entries of the rows of N as mentioned in Claim 33. Compute $A = \mathrm{diag}(M_{\sigma_0}, \ldots, M_{\sigma_{d-1}})$ and return black box access to $f(A^{-1} \cdot \mathbf{x})$

▶ **Definition 29.** *(List of nice polynomials in $\mathbb{F}_d[z]_k$): $\{h_0, \ldots, h_{d-k-1}\} \subseteq \mathbb{F}_d[z]_k$ is called a list of nice polynomials if the following properties are satisfied:*
1. *For distinct $r_1, r_2 \in [d-k]$, $h_{r_1}(\ell) = h_{r_2}(\ell)$ for every $\ell \in [k]$ and $h_{r_1}(\ell) \neq h_{r_2}(\ell)$ for every $\ell \in \{k, \ldots, d-1\}$.*
2. *For every $r \in [d-k], \sigma_0(h_r(0)), \ldots, \sigma_k(h_r(k))$ can be computed in poly(d, ρ) time.*

▷ Claim 30. A list of $d-k$ nice polynomials $\{h_0, \ldots, h_{d-k-1}\}$ can be computed in $\mathrm{poly}(d, \rho)$ time.
Using the list of nice polynomials, we recover $d-k$ distinct entries of $\sigma_0, \ldots, \sigma_{d-1}$.

▷ Claim 31. Given a list of nice polynomials $\{h_0, \ldots, h_{d-k-1}\}$, we can recover $d-k$ distinct entries in each of $\sigma_0, \ldots, \sigma_{d-1}$ in $\mathrm{poly}(d, \rho)$ time.

The matrix N defined in the algorithm is filled with some known entries and some unknowns. The goal is to recover all the entries of N which is accomplished by the following claims.

▷ Claim 32. Suppose $k \in [1, \frac{d}{3}]$. Then, there exist $k+1$ rows in N such that in each of these rows at least $k+1$ entries are known.

▷ Claim 33. Using $k+1$ rows of N indexed by $l_0, \ldots, l_k$ (as mentioned in Step 4), we can recover all the entries of N in $\mathrm{poly}(d, \rho)$ time.

6.1.2 Scaling equivalence test for NW

We present an algorithm for solving the scaling equivalence test for NW over a finite field $\mathbb{F}$, where $d \nmid |\mathbb{F}| - 1$. The same algorithm with appropriate modifications works over $\mathbb{R}$. More details on this are given in Section F.4 in [24]. Assume that f is scaling equivalent to NW.

Proof of correctness. The following claims and observations argue the correctness of the algorithm. The proofs of the claims are given in Section F.3 in [24].

▷ Claim 34. We can assume that $\alpha_{1,0} = \ldots = \alpha_{d-1,0} = 1$ without loss of generality.

The following observation can be proved easily.

▶ **Observation 6.2.** *Given a monomial m, we can recover the coefficient of m in f in* $\mathrm{poly}(d, \rho)$ *time.*

▷ Claim 35. In Step 4, $\alpha_{i,j}$ can be computed in $\mathrm{poly}(d, \rho)$ time. Further, $f = \mathsf{NW}(B \cdot \mathbf{x})$.

Algorithm 4 Scaling equivalence test for NW over finite fields.

Input: Black box access to $f \in \mathbb{F}[\mathbf{x}]$.
Output: An invertible diagonal matrix B such that $f = \mathsf{NW}(B \cdot \mathbf{x})$.

1. Let $B = \text{diag}(\alpha_{0,0}, \ldots, \alpha_{d-1,d-1})$, where $\{\alpha_{i,j} : i,j \in [d]\}$ are unknown. Set $\alpha_{1,0} = \ldots = \alpha_{d-1,0} = 1$ (Claim 34).
2. Let $S = (0, z, \ldots, (d-1)z,\ 1, z+1 \ldots, (d-1)z+1,\ \ldots\ , d-2, z+d-2 \ldots, (d-1)z+d-2,\ d-1)$ be the ordered set of d^2-d+1 polynomials in $\mathbb{F}[z]$. For every $h \in S$, query the coefficient c_h of the monomial $\prod_{i \in \mathbb{F}_d} x_{i,h(i)}$ from the black-box of f (Observation 6.2).
3. Let C be a $0/1$ matrix of size $(d^2-d+1) \times (d^2-d+1)$ whose rows and columns are indexed by S and $\mathbf{y} = (y_{0,0}, \ldots, y_{0,d-1}, y_{1,1}, \ldots, y_{1,d-1}, \ldots, y_{d-1,1}, \ldots, y_{d-1,d-1})$, respectively, such that for $h \in S$ and $y_{i,j} \in \mathbf{y}$, the $(h, y_{i,j})$-th entry of C is 1 if $h(i) = j$. (It is argued in Claim 39 in [24] that $|\det(C)|$ is a power of d). Compute the inverse of $\det(C)$ in $\mathbb{Z}_{|\mathbb{F}|-1}$ and denote it by γ. (Note that $\mathbf{y}$ does not contain the variables $\{y_{1,0}, \ldots, y_{d-1,0}\}$.)
4. Fix $\alpha_{i,j} \in \{\alpha_{0,0}, \ldots, \alpha_{d-1,d-1}\} \setminus \{\alpha_{1,0}, \ldots, \alpha_{d-1,0}\}$ arbitrarily. For every $h \in S$, compute the minor of C with respect to the row and column indexed by h and $y_{i,j}$ respectively and call it δ_h. Set $\alpha_{i,j} = \prod_{h \in S} c_h^{(\delta_h \cdot \gamma) \mod (|\mathbb{F}|-1)}$.
5. Set $B = \text{diag}(\alpha_{0,0}, \ldots, \alpha_{d-1,d-1})$. Return B. (see Claim 35)

7 Few problems

In conclusion, we state a few problems on the NW polynomial which, if resolved, would shed more light on this fundamental polynomial family.

1. Is the $\mathcal{NW} = \{\mathsf{NW}_{d,k} : d \text{ is a prime}\}$ family VNP-complete for a suitable choice of k (say, $k = d^{\epsilon}$ for a constant $\epsilon > 0$)?
2. Is there an efficient algorithm to check if $\mathsf{NW}(\mathbf{a}) = 0$ at a given point $\mathbf{a} \in \{0,1\}^n$? This problem was also posed in [6] [12].
3. Is there an efficient general equivalence test for NW? Theorem 4 may turn out to be a vital ingredient in such a test.
4. Give a complete description of the permutation symmetries of NW. Are all the permutation symmetries captured in Lemma 45 mentioned in Section D in [24]?

For the permanent polynomial, the solutions to these problems are well known.

References

1 Scott Aaronson. P=?NP. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:4, 2017.

2 Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

3 Noga Alon and Ravi B. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987.

4 Albert Atserias. Distinguishing SAT from Polynomial-Size Circuits, through Black-Box Queries. In *21st Annual IEEE Conference on Computational Complexity (CCC 2006), 16-20 July 2006, Prague, Czech Republic*, pages 88–95, 2006.

[12] We thank Andrej Bogdanov for pointing this out to us.

5 Markus Bläser, Christian Ikenmeyer, Gorav Jindal, and Vladimir Lysikov. Generalized matrix completion and algebraic natural proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1193–1206, 2018.

6 Andrej Bogdanov and Muli Safra. Hardness Amplification for Errorless Heuristics. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 418–426, 2007.

7 Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No Occurrence Obstructions in Geometric Complexity Theory. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 386–395, 2016.

8 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.

9 Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth Multilinear Formula Lower Bounds for Iterated Matrix Multiplication, with Applications. In *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, pages 21:1–21:15, 2018.

10 Suryajith Chillara and Partha Mukhopadhyay. Depth-4 Lower Bounds, Determinantal Complexity: A Unified Approach. In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014), STACS 2014, March 5-8, 2014, Lyon, France*, pages 239–250, 2014.

11 Klim Efremenko, Ankit Garg, Rafael Mendes de Oliveira, and Avi Wigderson. Barriers for Rank Methods in Arithmetic Complexity. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, pages 1:1–1:19, 2018.

12 Michael A. Forbes, Mrinal Kumar, and Ramprasad Saptharishi. Functional Lower Bounds for Arithmetic Circuits and Connections to Boolean Circuit Complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 33:1–33:19, 2016.

13 Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving algebraic circuits lower bounds. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 653–664, 2017.

14 Lance Fortnow, Aduri Pavan, and Samik Sengupta. Proving SAT does not have small circuits with an application to the two queries problem. *J. Comput. Syst. Sci.*, 74(3):358–363, 2008.

15 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower Bounds for Depth-4 Formulas Computing Iterated Matrix Multiplication. *SIAM J. Comput.*, 44(5):1173–1201, 2015.

16 Georg Frobenius. Ueber die darstellung der endlichen gruppen durch linearc substitutionen. *Sitzungber. der Berliner Akademie*, 7:994–1015, 1897.

17 Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, Circuits, and the Polynomial-Time Hierarchy. In *22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981*, pages 260–270, 1981.

18 Ankit Garg, Nikhil Gupta, Neeraj Kayal, and Chandan Saha. Determinant equivalence test over finite fields and over Q. *Electronic Colloquium on Computational Complexity (ECCC)*, 26:42, 2019.

19 Fulvio Gesmundo. Gemetric aspects of iterated matrix multiplication. *Journal of Algebra*, 461:42–64, 2016.

20 Dima Grigoriev and Marek Karpinski. An Exponential Lower Bound for Depth 3 Arithmetic Circuits. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 577–582, 1998.

21 Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017.

22 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.

23 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *J. ACM*, 61(6):33:1–33:16, 2014.

24 Nikhil Gupta and Chandan Saha. On the symmetries of and equivalence test for design polynomials. *ECCC*, 2019. URL: `https://eccc.weizmann.ac.il/report/2018/164/`.

25 Brian C Hall. *Lie Groups, Lie Algebras and Representations An Elementary introduction.* Springer, second edition, 2015.

26 Johan Håstad. Almost Optimal Lower Bounds for Small Depth Circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20, 1986.

27 Jesko Hüttenhain. The Stabilizer of Elementary Symmetric Polynomials. *CoRR*, abs/1607.08419, 2016. URL: `https://arxiv.org/abs/1607.08419`.

28 Christian Ikenmeyer, Ketan D. Mulmuley, and Michael Walter. On vanishing of Kronecker coefficients. *Computational Complexity*, 26(4):949–992, 2017.

29 Christian Ikenmeyer and Greta Panova. Rectangular Kronecker Coefficients and Plethysms in Geometric Complexity Theory. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 396–405, 2016.

30 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.

31 Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Formulas. *SIAM J. Comput.*, 46(1):307–335, 2017.

32 Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of Full Rank Algebraic Branching Programs. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 21:1–21:61, 2017.

33 Neeraj Kayal and Chandan Saha. Lower Bounds for Depth-Three Arithmetic Circuits with small bottom fanin. *Computational Complexity*, 25(2):419–454, 2016.

34 Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 146–153, 2014.

35 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An Almost Cubic Lower Bound for Depth Three Arithmetic Circuits. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 33:1–33:15, 2016.

36 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth four formulas with low individual degree. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 626–632, 2016.

37 Mrinal Kumar and Ramprasad Saptharishi. An Exponential Lower Bound for Homogeneous Depth-5 Circuits over Finite Fields. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 31:1–31:30, 2017.

38 Mrinal Kumar and Shubhangi Saraf. Superpolynomial Lower Bounds for General Homogeneous Depth 4 Arithmetic Circuits. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 751–762, 2014.

39 Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: it's all about the top fan-in. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 136–145, 2014.

40 Mrinal Kumar and Shubhangi Saraf. Arithmetic Circuits with Locally Low Algebraic Rank. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 34:1–34:27, 2016.

41 Mrinal Kumar and Shubhangi Saraf. Sums of Products of Polynomials in Few Variables: Lower Bounds and Polynomial Identity Testing. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 35:1–35:29, 2016.

42 Mrinal Kumar and Shubhangi Saraf. On the Power of Homogeneous Depth 4 Arithmetic Circuits. *SIAM J. Comput.*, 46(1):336–387, 2017.

43 Richard J. Lipton. New Directions In Testing. In *Distributed Computing And Cryptography, Proceedings of a DIMACS Workshop, Princeton, New Jersey, USA, October 4-6, 1989*, pages 191–202, 1989.

44 Marvin Marcus and Francis May. The permanent function. *Canadian Journal of Mathematics*, 14:177–189, 1962.

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

46 Ketan Mulmuley. Lower Bounds in a Parallel Model without Bit Operations. *SIAM J. Comput.*, 28(4):1460–1509, 1999.

47 Ketan Mulmuley. On P vs. NP, Geometric Complexity Theory, and the Flip I: a high level view. *CoRR*, abs/0709.0748, 2007.

48 Ketan Mulmuley. Explicit Proofs and The Flip. *CoRR*, abs/1009.0246, 2010. URL: `http://arxiv.org/abs/1009.0246`.

49 Ketan Mulmuley. On P vs. NP and geometric complexity theory: Dedicated to Sri Ramakrishna. *J. ACM*, 58(2):5:1–5:26, 2011.

50 Ketan Mulmuley. The GCT program toward the *P* vs. *NP* problem. *Commun. ACM*, 55(6):98–107, 2012.

51 Ketan Mulmuley and Milind A. Sohoni. Geometric Complexity Theory I: An Approach to the P vs. NP and Related Problems. *SIAM J. Comput.*, 31(2):496–526, 2001.

52 Cody Murray and R. Ryan Williams. Circuit lower bounds for nondeterministic quasi-polytime: an easy witness lemma for NP and NQP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 890–901, 2018.

53 Noam Nisan and Avi Wigderson. Hardness vs Randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

54 Noam Nisan and Avi Wigderson. Lower Bounds on Arithmetic Circuits Via Partial Derivatives. *Computational Complexity*, 6(3):217–234, 1997.

55 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009.

56 Ran Raz and Amir Yehudayoff. Lower Bounds and Separations for Constant Depth Multilinear Circuits. *Computational Complexity*, 18(2):171–207, 2009.

57 Alexander A. Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Soviet Mathematics Doklady*, 31:354–357, 1985.

58 Alexander A. Razborov. Lower bounds on the size of bounded-depth networks over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

59 Alexander A. Razborov and Steven Rudich. Natural Proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

60 Kenneth W. Regan. Understanding the Mulmuley-Sohoni Approach to P vs. NP. *Bulletin of the EATCS*, 78:86–99, 2002.

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

62 Roman Smolensky. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 77–82, 1987.

63 Ryan Williams. Nonuniform ACC Circuit Lower Bounds. *J. ACM*, 61(1):2:1–2:32, 2014.

64 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.