# On the symmetries of and equivalence test for design polynomials

Nikhil Gupta, Indian Institute of Science

(Joint work with Chandan Saha)

# Overview

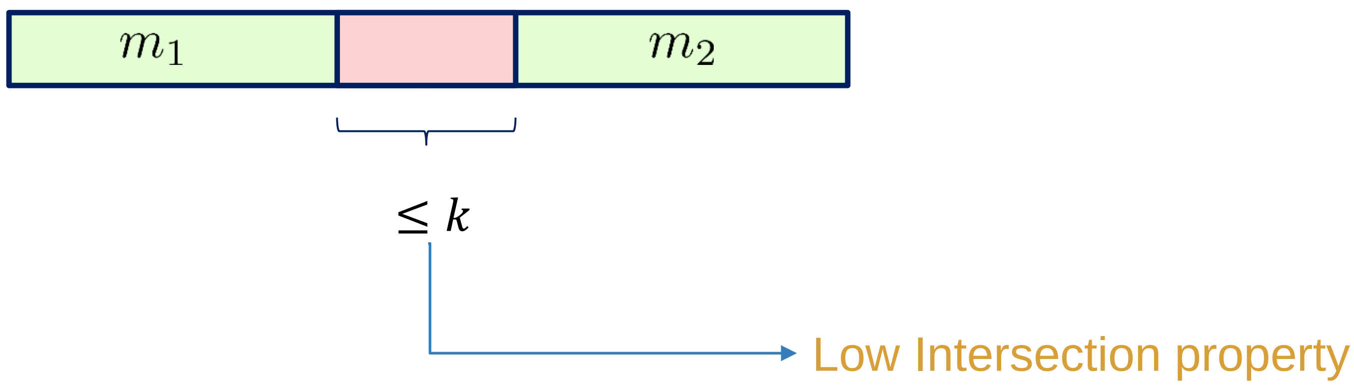
Introduction to design polynomials → Some natural questions about them → Our results

# 1.

# Introduction to design polynomials

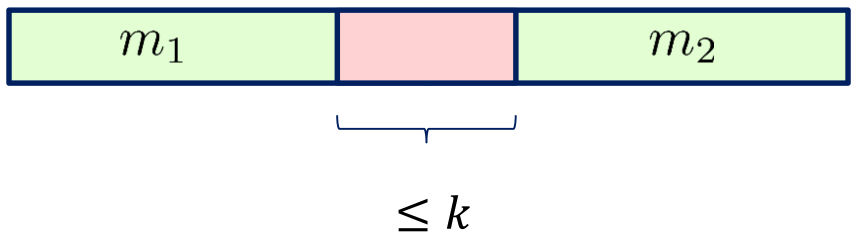
# Introduction

- Design polynomial: It is a sum of multilinear monomials, where each pair of monomials has low overlap.



# Introduction

- **Design polynomial**: It is a sum of multilinear monomials, where each pair of monomials has low overlap.




- **Nisan-Wigderson design polynomial**: Let
  - $\mathrm{F}_d = \{1, \dots, d\}$ be a prime finite field
  - $\mathrm{F}_d[z]_k = \{h \in \mathrm{F}_d[z] \mid \deg(h) \leq k\}, \mathbf{x} = \{x_{1,1}, \dots, x_{d,d}\}$

$$\mathrm{NW}(\mathbf{x}) = \sum_{h \in \mathrm{F}_d[z]_k} x_{1,h(1)} \cdots x_{d,h(d)}$$

# Introduction

- NW is a homogeneous polynomial of degree $d$ having $d^{k+1}$ monomials.

- NW is a set multilinear polynomial w.r.t

$$\mathbf{x} = \mathbf{x}_1 \uplus \cdots \uplus \mathbf{x}_d,$$

where $\mathbf{x}_i = \{x_{i,j} \mid j \in \{1, \dots, d\}\}$.

- In a set-multilinear polynomial , every monomial has exactly 1 variable from each $\mathbf{x}_i$

# Introduction

- Inspired from the Nisan-Wigderson design.
  - used in hardness-randomness trade off.
- NW polynomial was introduced in [KSS14].

*A super-polynomial lower bound for regular arithmetic formulas*
by Kayal, Saha and Saptharishi

# Introduction

- Inspired from the Nisan-Wigderson design.
  - used in hardness-randomness trade off.
- NW polynomial was introduced in [KSS14] .
- NW polynomial has been used to prove lower bound on the size of several classes of arithmetic circuits.
- Permanent, Determinant, Iterated matrix multiplication (IMM) etc have also be used to prove lower bounds.

# Introduction

- The other polynomials are well studied .
- But unlike these, very little is known about NW.
  - $\mathrm{NW} \in \mathrm{VNP}$.

# Introduction

- The other polynomials are well studied .
- But unlike these, very little is known about NW.
  - $\mathrm{NW} \in \mathrm{VNP}$.
- We aim to study some natural questions about NW.
- We would be comparing NW with the Permanent in this talk. Recall,

$$\mathrm{Perm_d}(\mathbf{x}) = \sum_{\text{Permutation } \sigma \text{ on } [d]} x_{1,\sigma(1)} \cdots x_{d,\sigma(d)}$$

# 2.

# Some natural questions about a polynomial family

Structural Questions

Algorithmic Questions

Complexity theoretic Questions

# Structural aspects: Symmetries of a polynomial

- An invertible matrix B is called a symmetry of a polynomial $f \in \mathbb{F}[\mathbf{x}]$ if
$$f(B \cdot \mathbf{x}) = f(\mathbf{x}).$$

# Structural aspects: Symmetries of a polynomial

- An invertible matrix B is called a symmetry of a polynomial $f \in \mathrm{F}[\mathbf{x}]$ if
$$f(B \cdot \mathbf{x}) = f(\mathbf{x}).$$

- The set of symmetries of $f$ is a group w.r.t matrix multiplication, denoted $G_f$.

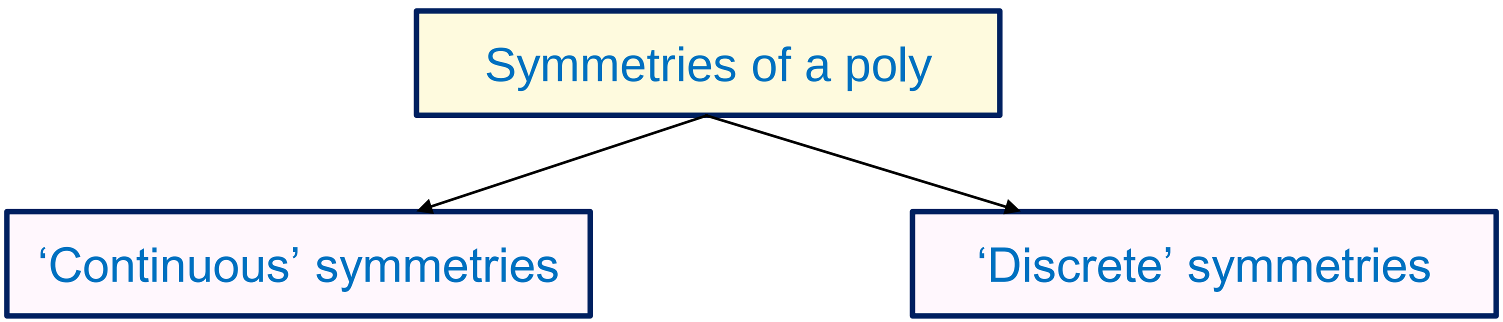
- Example:
$f = x_1^2 + x_2^2$ , $B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Then, $f(B \cdot \mathbf{x}) = f(\mathbf{x})$.

# Symmetries of a polynomial

- An invertible matrix B is called a symmetry of a polynomial $f \in \mathbb{F}[\mathbf{x}]$ if

$$f(B \cdot \mathbf{x}) = f(\mathbf{x}).$$



# Symmetries of a polynomial

- An invertible matrix B is called a symmetry of a polynomial $f \in \mathbb{F}[\mathbf{x}]$ if

$$f(B \cdot \mathbf{x}) = f(\mathbf{x}).$$

Symmetries of a poly

‘Continuous’ symmetries

‘Discrete’ symmetries

Example: $f = x_1^2 + x_2^2, t \in [0,2\pi]$

$$B_t = \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix}$$

$$f(B_t \cdot \mathbf{x}) = (x_1^2+x_2^2)(\sin^2 t + cos^2 t) = f$$

Example: $f = x_1^2 + x_2^2$

$$B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$f(B \cdot \mathbf{x}) = f$$

# Symmetries of a polynomial

- **Question:** Why are the symmetries interesting?

# Symmetries of a polynomial

- Question: Why are the symmetries interesting?

- Naturally interesting in invariant theory.

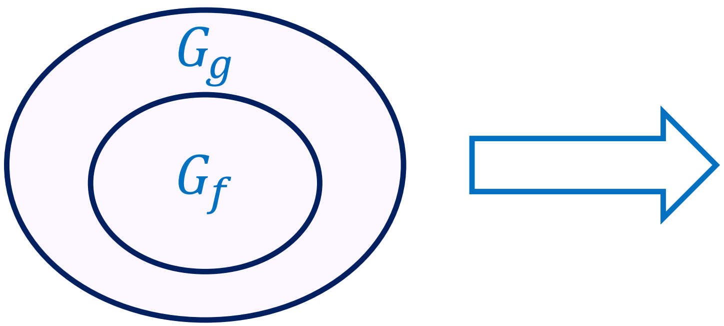
- Helps in the equivalence test of a polynomial.

- Possible that this knowledge could help in understanding the complexity of the underlying polynomial family (the GCT approach).

- GCT aims to separate VP from VNP by exploiting the characterization by symmetries property of the Permanent and the Determinant.

# Characterization by symmetries

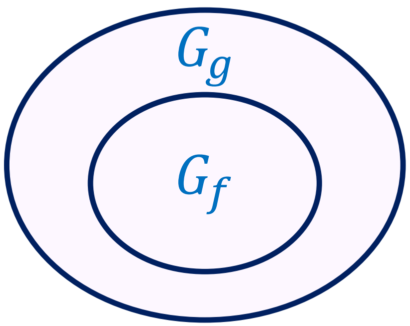
- Let $f, g \in \mathbb{F}[\mathbf{x}]$ be degree $d$ homogeneous polynomials. Then, $f$ is characterized by its symmetries over $\mathbb{F}$ if




$g = \alpha \cdot f$, for some $\alpha \in \mathbb{F}$

# Characterization by symmetries

- Let $f, g \in \mathbb{F}[\mathbf{x}]$ be degree $d$ homogeneous polynomials. Then, $f$ is characterized by its symmetries over $\mathbb{F}$ if



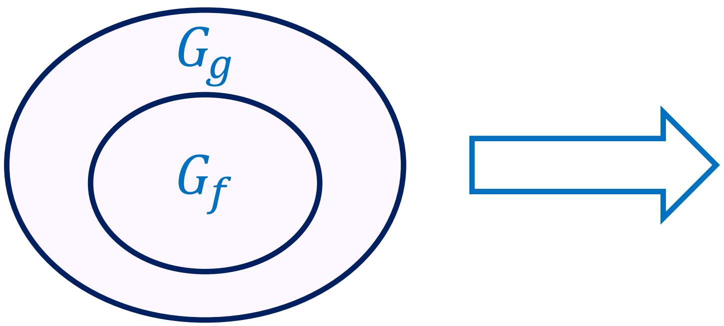

$g = \alpha \cdot f$, for some $\alpha \in \mathbb{F}$

- A random polynomial is not characterized by its symmetries [Gro12].

Ph.D. thesis of
Joshua A. Grochow

# Characterization by symmetries

- Let $f, g \in \mathbb{F}[\mathbf{x}]$ be degree $d$ homogeneous polynomials. Then, $f$ is characterized by its symmetries over $\mathbb{F}$ if




$g = \alpha \cdot f$ , for some $\alpha \in \mathbb{F}$

- A random polynomial is not characterized by its symmetries [Gro12].

- It is a property satisfied by a 'very small fraction' of polynomials [Gro12].

# Some natural structural questions

| | |
|---|---|
| Question 1 | Is a polynomial characterized by its symmetries over **C,R** and finite fields? |
| Question 2 | What is the structure of the group of symmetries of a polynomial *f* ? |
| Question 3 | Given a polynomial *f*, does there exist an algorithm that determines $G_f$ ?<br>• Known for 'binary forms' over **C** and **R** [BO00]. |

*Symmetries of polynomials* by Berchenko and Olver

# Some natural structural questions

| | |
|---|---|
| Question 1 | Is a polynomial characterized by its symmetries over **R**,**C** and finite fields? |
| Question 2 | What is the structure of the group of symmetries of a polynomial $f$ ? |
| Question 3 | Given a polynomial $f$, does there exist an algorithm that determines $G_f$ ?<br>• Known for 'binary forms' over **C** and **R** [BO00]. |
| Question 4 | Given a polynomial $f$ as a list of coefficients over a field F. Can we test efficiently if $f$ is characterized by its symmetries?<br>• Open problem stated in [Gro12] |

# Comparison

| Question | Permanent | NW (Our results) |
|---|---|---|
| 1. Characterized by its symmetries? | Yes (over almost all the fields) | • Yes (Over **C**)<br>• No (Over **R**) |

# Comparison

| Question | Permanent | NW (Our results) |
|---|---|---|
| 1. Characterized by its symmetries? | Yes (over almost all the fields) | • Yes (Over **C**)<br>• No (Over **R**) |
| 2. Is every symmetry a product of permutation and diagonal matrices? | Yes | Yes |

# Comparison

| Question | Permanent | NW (Our results) |
|---|---|---|
| 1. Characterized by its symmetries? | Yes (over almost all the fields) | • Yes (Over **C**)<br>• No (Over **R**) |
| 2. Is every symmetry a product of permutation and diagonal matrices? | Yes | Yes |
| 2.a. Are all the diagonal symmetries known? | Yes | • Yes (Over **R**)<br>• Over **C** too. |

# Comparison

| Question | Permanent | NW (Our results) |
|---|---|---|
| 1. Characterized by its symmetries? | Yes (over almost all the fields) | • Yes (Over **C**)<br>• No (Over **R**) |
| 2. Is every symmetry a product of permutation and diagonal matrices? | Yes | Yes |
| 2.a. Are all the diagonal symmetries known? | Yes | • Yes (Over **R**)<br>• Over **C** too |
| 2.b. Are all the permutation symmetries known? | Yes | Partially |

# Comparison

| Question | Permanent | NW (Our results) |
|---|---|---|
| 1. Characterized by its symmetries? | Yes (over almost all the fields) | • Yes (Over **C**)<br>• No (Over **R**) |
| 2. Is every symmetry a product of permutation and diagonal matrices? | Yes | Yes |
| 2.a. Are all the diagonal symmetries known? | Yes | • Yes (Over **R**)<br>• Over **C** too |
| 2.b. Are all the permutation symmetries known? | Yes | Partially |
| 2.c. Are all the diagonal symmetries 'continuous'? | Yes | • No (Over **C**)<br>• Yes (Over **R**) |

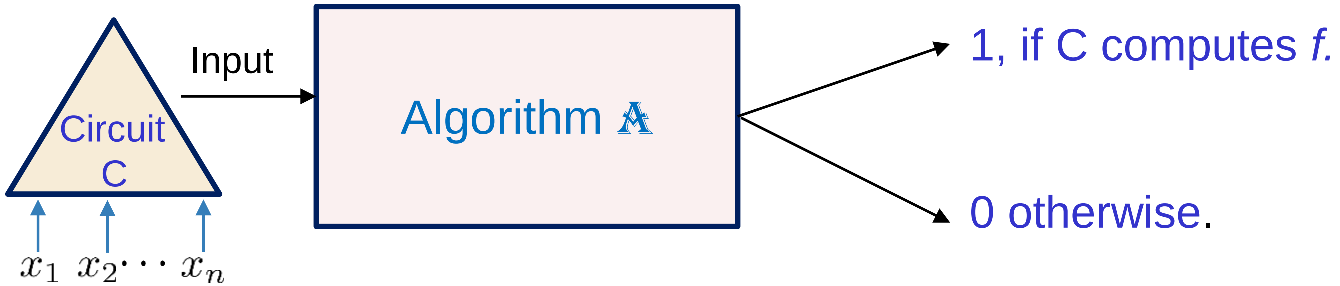
# Comparison

| Question | Permanent | NW (Our results) |
|---|---|---|
| 1. Characterized by its symmetries? | Yes (over almost all the fields) | • Yes (Over **C**)<br>• No (Over **R**) |
| 2. Is every symmetry a product of permutation and diagonal matrices? | Yes | Yes |
| 2.a. Are all the diagonal symmetries known? | Yes | • Yes (Over **R**)<br>• Over **C** too |
| 2.b. Are all the permutation symmetries known? | Yes | Partially |
| 2.c. Are all the diagonal symmetries 'continuous'? | Yes | • No (Over **C**)<br>• Yes (Over **R**) |

Helps in symmetry characterization

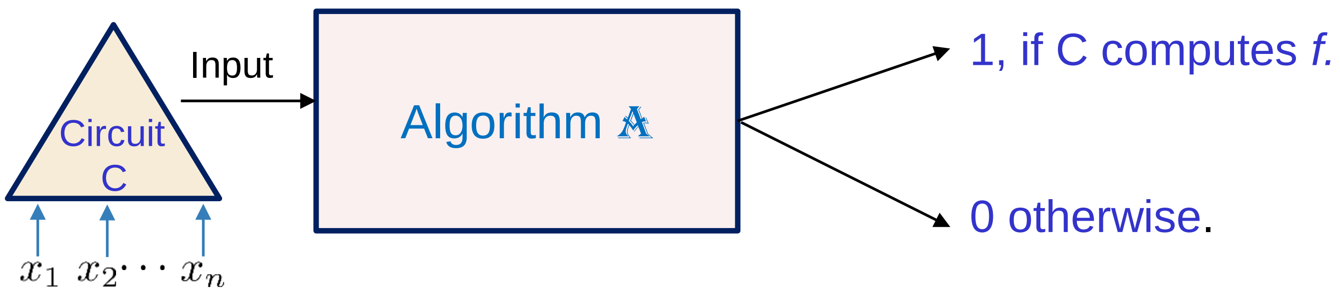
# Algorithmic aspects: 1. Circuit Testing

- Let $f \in \mathbb{F}[\mathbf{x}]$. Does there exist an algorithm $\mathbb{A}$, s.t.



# Algorithmic aspects: 1. Circuit Testing

▷ Let $f \in \mathbb{F}[\mathbf{x}]$. Does there exist an algorithm $\mathcal{A}$ , s.t.

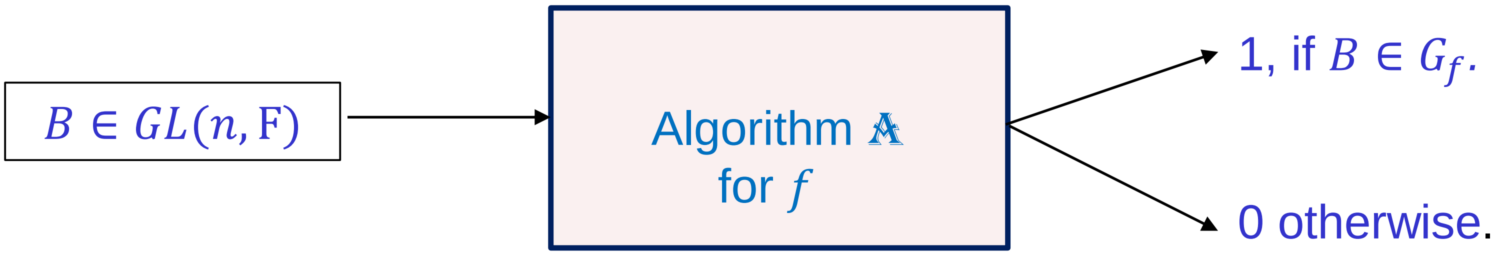



▷ Interesting when it is not known if $f$ is computed by a circuit of small size.

▷ Knowledge of symmetries of $f$ can be helpful.

# 2. Symmetry Testing

- Does there exist an algorithm $\mathfrak{A}$, such that



# 2. Symmetry Testing

- Does there exist an algorithm $\mathfrak{A}$, such that


# 3. Equivalence Test

- $f, g \in \mathrm{F}[\mathbf{x}]$ are called equivalent if there exists an invertible matrix $B$ over a field F, such that

$$f(\mathbf{x}) = g(B \cdot \mathbf{x}).$$

# 3. Equivalence Test

- **Equivalence test**: Let $f \in \mathbb{F}[\mathbf{x}]$. Does there exist an algorithm $\mathbb{A}$, such that

# 3. Equivalence Test

- **Equivalence test**: Let $f \in \mathbb{F}[\mathbf{x}]$. Does there exist an algorithm $\mathfrak{A}$, such that

Black box of $g \in \mathbb{F}[\mathbf{x}]$ → Algorithm $\mathfrak{A}$: Is $g$ equivalent to $f$ ?

Yes → $B$, s.t. $g = f(B \cdot \mathbf{x})$

No → 0 otherwise.

- Equivalence test for arbitrary $f$ and $g$ is at least as hard as graph isomorphism [AS04].

*Equivalence of F-Algebras and Cubic Forms by* Agarwal and Saxena

# 3. Equivalence Test

- Equivalence test: Let $f \in \mathbb{F}[\mathbf{x}]$. Does there exist an algorithm $\mathfrak{A}$, such that

Black box of $g \in \mathbb{F}[\mathbf{x}]$ → Algorithm $\mathfrak{A}$: Is $g$ equivalent to $f$?

Yes → $B$, s.t. $g = f(B \cdot \mathbf{x})$

No → 0 otherwise.

- Equivalence test for arbitrary $f$ and $g$ is at least as hard as graph isomorphism [AS04].

*Affine projections of polynomials* by Kayal

*Reconstruction of full rank ABP* by Kayal, Nair, Saha and Tavenas

- It was shown in [Kay12] and [KNST17] that if $f$ is fixed to Permanent, Determinant, IMM etc then equivalence test (over **C**) is solved in randomized poly time.

# 4. Flip Theorem for $\{f_n\}$

- Suppose $f_n$ is not computed by circuits of size $\leq s$. Then, $\exists$ $\{\alpha_1, \dots, \alpha_m\} \subseteq \mathrm{F}^n, m = \mathrm{poly}(n)$, such that
  $\forall$ size $\leq s$ circuit $C, \exists\, \ell \in [m], C(\alpha_\ell) \neq f(\alpha_\ell)$.

# 4. Flip Theorem for $\{f_n\}$

- Suppose $f_n$ is not computed by circuits of size $\leq s$.
  Then, $\exists$ $\{\alpha_1, \dots, \alpha_m\} \subseteq \mathrm{F}^n, m = \mathrm{poly}(n)$, such that
  $\forall$ size $\leq s$ circuit $C, \exists\, \ell \in [m], C(\alpha_\ell) \neq f(\alpha_\ell)$.

# 4. Flip Theorem for $\{f_n\}$

- Suppose $f_n$ is not computed by circuits of size $\leq s$. Then, $\exists$ $\{\alpha_1, \dots, \alpha_m\} \subseteq \mathrm{F}^n, m = \mathrm{poly}(n)$, such that
  $\forall$ size $\leq s$ circuit $C, \exists \ell \in [m], C(\alpha_\ell) \neq f(\alpha_\ell)$.


- A version of Flip theorem is known for SAT [FPS08]

Proving SAT does not have small circuits with an application to the two queries problem by *Fortnow, Pavan and Sengupta*

# 4. Flip Theorem for $\{f_n\}$

- Suppose $f_n$ is not computed by circuits of size $\leq s$. Then, $\exists$ $\{\alpha_1, \dots, \alpha_m\} \subseteq \mathbb{F}^n, m = \mathrm{poly}(n)$, such that
  $\forall$ size $\leq s$ circuit $C, \exists\, \ell \in [m], C(\alpha_\ell) \neq f(\alpha_\ell)$.


- A version of Flip theorem is known for SAT [FPS08]

- Known for the Permanent[Mul10]

Explicit proofs and the flip by *Mulmuley*

# 5. Zero Testing on $\{0,1\}^n$

- Let $f \in \mathbb{F}[\mathbf{x}]$ be an n variate polynomial. Does there exist an algorithm $\mathfrak{A}$, such that

# 5. Zero Testing on $\{0,1\}^n$

- Let $f \in \mathbb{F}[\mathbf{x}]$ be an n variate polynomial. Does there exist an algorithm $\mathbb{A}$, such that

$\boldsymbol{a} \in \{0,1\}^n$ → Algorithm $\mathbb{A}$ for $f$ → $1,\ f(\boldsymbol{a}) \neq 0$ / 0 otherwise.

- It may shed some light on the circuit complexity of $f$.

- In case of NW, this problem is also relevant for hardness amplification [BS07].

*Hardness amplification for errorless heuristic* by Bogdanov and Safra

# Comparison

| Question | Permanent | NW (Our results) |
|---|---|---|
| 1. Circuit Testing | Yes | Yes |
| 2. Symmetry Testing | Yes | • Yes for diagonal matrices<br>• Partially for permutation matrices |
| 3. Equivalence Test | Yes | Partially Over **R** and $\mathbb{F}_p$<br>• Certain PS matrices |
| 4. Flip theorem | Yes | Yes |
| 5. Zero Testing | Yes | Open problem mentioned in [BS07]. |

❑ These algorithmic results crucially use the knowledge of $G_{NW}$.

# Complexity theoretic aspects: VNP completeness and circuit compression

- Is a given polynomial VNP complete?

- Circuit Compression: Let $f$ be a polynomial having $m$ monomials. Can $f$ be computed by a circuit of size significantly less than $m$?

| Question | Permanent | NW |
|---|---|---|
| 1. VNP completeness | Yes | OPEN |
| 2. Circuit compression | Yes<br>Computed by a circuit of size $2^{O(n)}$ | OPEN |

# 3.

# Proofs of some results for NW

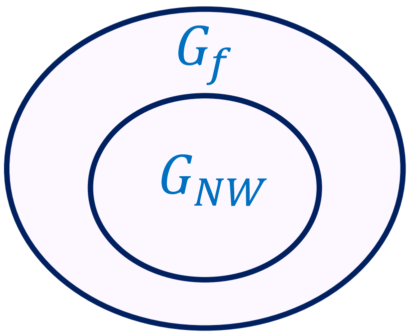
Symmetry characterization

Structure of $G_{NW}$

Special cases of equivalence test for NW

# 1. Symmetry characterization over **C**

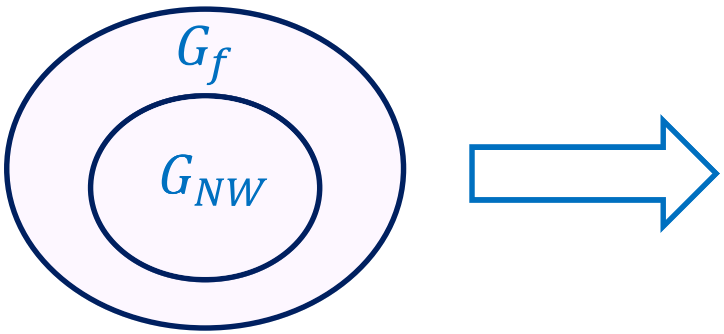
- **Theorem 1** : Let $f \in \mathbf{C}[\mathbf{x}]$ be a homogeneous degree $d$ polynomial. Then,




$\Rightarrow$ $f = \alpha \cdot \mathrm{NW}$, for some $\alpha \in \mathbf{C}$.

# 1. Symmetry characterization over $\mathbf{C}$

- Theorem 1 : Let $f \in \mathbf{C}[\mathbf{x}]$ be a homogeneous degree $d$ polynomial. Then,




$f = \alpha \cdot \mathrm{NW}$, for some $\alpha \in \mathbf{C}$.

- Recall, if $B \in G_{NW}$ then $\mathrm{NW}(B \cdot \mathbf{x}) = \mathrm{NW}$ and

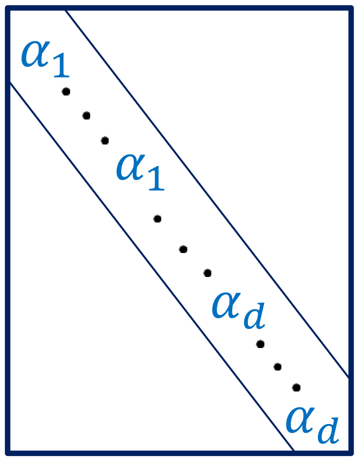
$$\mathrm{NW}(\mathbf{x}) = \sum_{h \in \mathrm{F}_d[z]_k} x_{1,h(1)} \cdots x_{d,h(d)}$$

$$\mathrm{F}_d[z]_k = \{h \in \mathrm{F}_d[z] \mid \deg(h) \leq k\}.$$

# 1. Symmetry characterization over **C**

▷ Some symmetries of NW over **C**:

1. Continuous diagonal symmetries:

$B =$ 

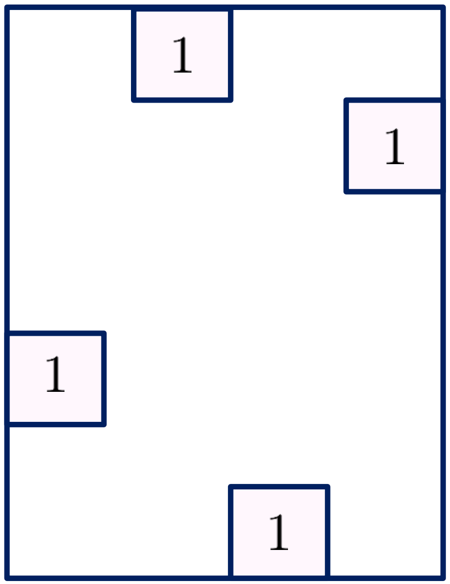

- $\alpha_i \in \mathbf{C}^\times, i \in \{1, \dots, d\}$.
- $\alpha_1 \cdots \alpha_d = 1$ .
- Each $\alpha_i$ appears exactly d times on the diagonal.

$$x_{1,h(1)} \cdots x_{d,h(d)} \xrightarrow{B \cdot \mathbf{x}} x_{1,h(1)} \cdots x_{d,h(d)}$$

# 1. Symmetry characterization over **C**

## 2. Permutation symmetries:

$B_q =$



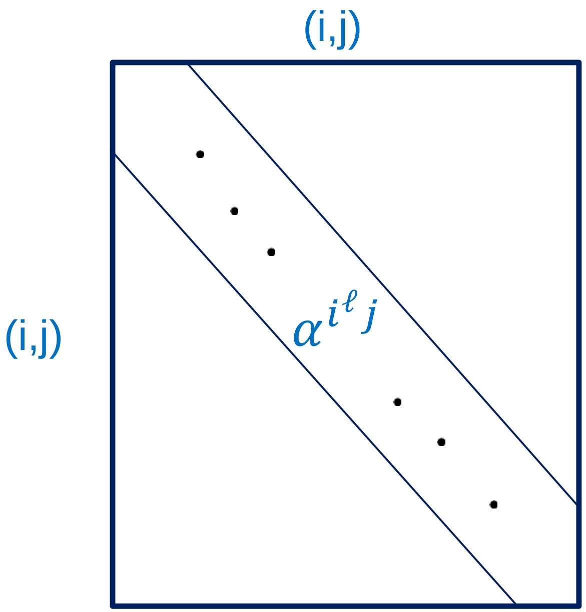

- For $q \in \mathrm{F}_d\,[z]_k, B_q$ maps $x_{i,j}$ to $x_{i,j+q(i)}$
- $B_q$ is a permutation matrix

$$x_{1,h(1)} \cdots x_{d,h(d)} \xrightarrow{B_q \cdot \mathbf{x}} x_{1,(h+q)(1)} \cdots x_{d,(h+q)(d)}$$

# 1. Symmetry characterization over **C**

## 3. Discrete diagonal symmetries:

$B_\ell =$




- $\alpha$ is the $d$ -th primitive root of unity.
- $\ell \in \{0, \dots, d-k-2\}$.
- For $h \in F_d[z]_k, \ell \in \{0, \dots, d-k-2\}$, $\prod_{i \in F_d} \alpha^{i^\ell \cdot h(i)} = 1$

$$x_{1,h(1)} \cdots x_{d,h(d)} \xrightarrow{B_\ell \cdot \mathbf{x}} \left(\prod_{i \in F_d} \alpha^{i^\ell \cdot h(i)}\right) x_{1,h(1)} \cdots x_{d,h(d)}$$

# 1. Symmetry characterization over **C**

▷ Proof of Theorem 1 : Suppose $G_{NW} \subseteq G_f$. Then,

Continuous diagonal symmetries $\Longrightarrow$ $f$ is set multilinear.

Permutation symmetries and discrete diagonal symmetries $\Longrightarrow$ $f = \alpha \cdot \mathrm{NW}$, for $\alpha \in \mathbf{C}$.

This completes the proof.

# 1. Symmetry characterization over **C**

- Proof of Theorem 1 : Suppose $G_{NW} \subseteq G_f$. Then,

Continuous diagonal symmetries $\Rightarrow$ $f$ is set multilinear.

Permutation symmetries and discrete diagonal symmetries $\Rightarrow$ $f = \alpha \cdot \mathrm{NW}$, for $\alpha \in \mathbf{C}$.

This completes the proof.

- Theorem : NW is not characterized by its symmetries over **R.**

# 2. Structure of $G_{NW}$

- Theorem 2: Let $B \in G_{NW}$. Then,

$$B = D \cdot P,$$

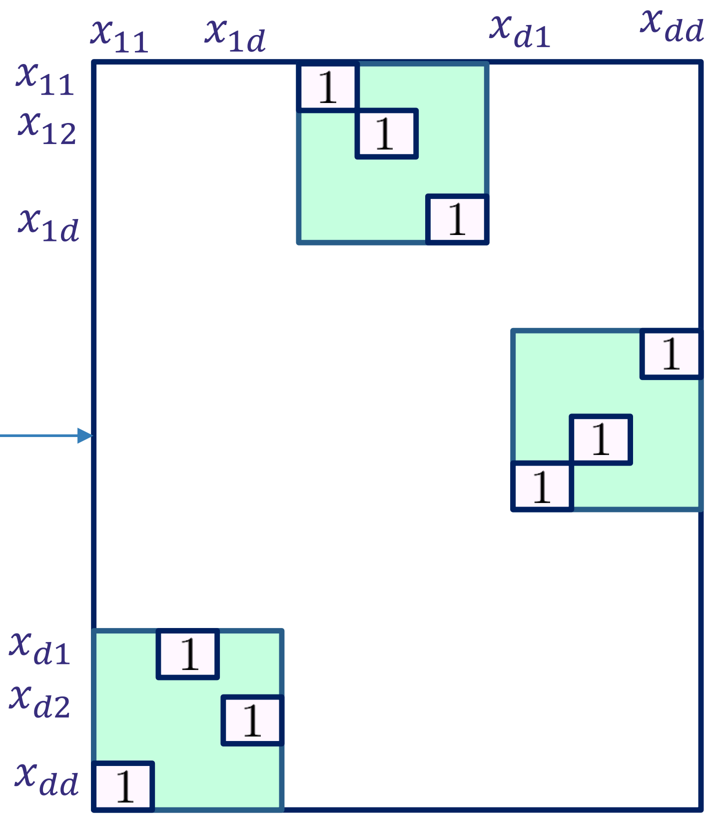
  - D is an invertible diagonal matrix
  - P is a block permuted permutation matrix

# 2. Structure of $G_{NW}$

- **Theorem 2**: Let $B \in G_{NW}$. Then,

$$B = D \cdot P,$$

  - D is an invertible diagonal matrix
  - P is a block permuted permutation matrix



# 2. Structure of $G_{NW}$

- Proof idea:

Step 1. Using the Lie algebra of NW, we show that every $B \in G_{NW}$ is a block permuted matrix.

↓

Step 2. Using the Hessian matrix and evaluation dimension, we show that $B = D \cdot P$.

# 2.a Lie algebra of a polynomial

- A matrix $A = \left(a_{ij}\right)_{i,j\in[n]}$ is in the Lie algebra of $f$, denoted $g_f$, if the following equation holds:

$$\sum_{i,j\in[n]} a_{ij} \cdot x_j \cdot \frac{\partial f}{\partial x_i} = 0$$

# 2.a Lie algebra of a polynomial

- A matrix $A = (a_{ij})_{i,j\in[n]}$ is in the Lie algebra of *f*, denoted $g_f$, if the following equation holds:

$$\sum_{i,j\in[n]} a_{ij} \cdot x_j \cdot \frac{\partial f}{\partial x_i} = 0$$

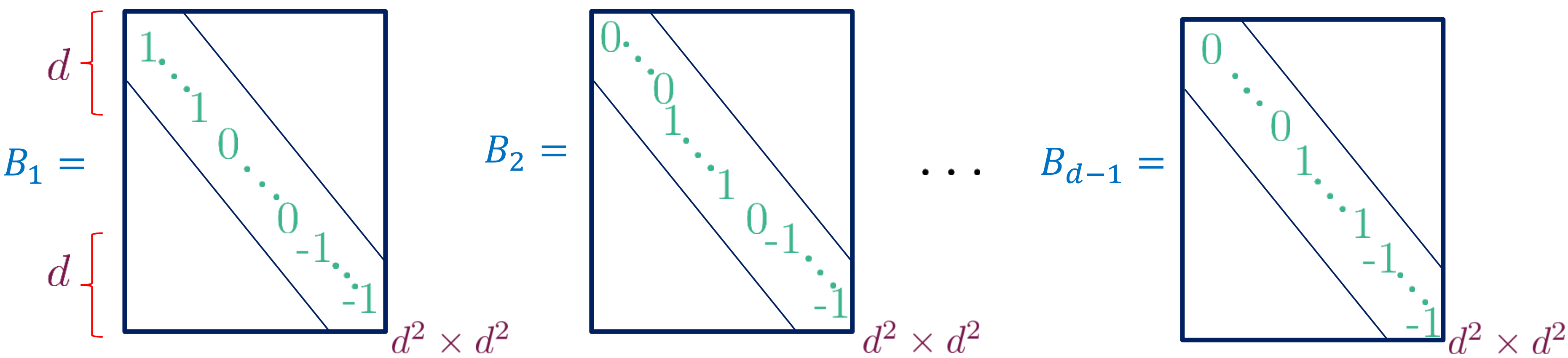
- $g_f$ is a vector space over F.

- Continuous symmetries of *f* are obtained from $g_f$.

$$\begin{aligned} exp: \quad g_f &\longrightarrow G_f \\ A &\longrightarrow e^A \end{aligned}$$

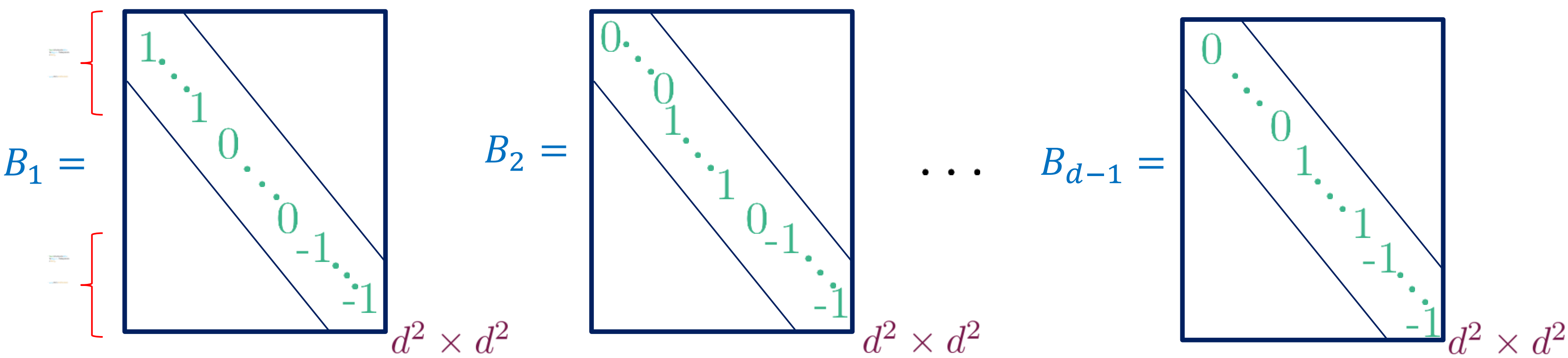
# 2.a Lie algebra of NW

- **Theorem 3:** Let F be a field, such that char(F) $\neq d$. Then, $\mathrm{Dim}(g_{NW}) = d - 1$. The following matrices form an F-basis of $g_{NW}$.



# 2.a Lie algebra of NW

- **Theorem 3:** Let F be a field, such that $\mathrm{char}(F) \neq d$. Then, $\mathrm{Dim}(g_{NW}) = d - 1$. The following matrices form an F-basis of $g_{NW}$.




- $g_{NW} \subseteq g_f$, where $f$ is a set-multilinear polynomial.

# 2.a Lie algebra of NW

- Proof idea:
  At the heart of the proof lies an understanding of the following system of linear equations:

  - Consider the following equations in the formal variables $\{\gamma_{i,j} \mid i,j \in \mathrm{F}_d\}$ for every $h \in \mathrm{F}_d[z]_k$:

$$\gamma_{1,h(1)} + \cdots + \gamma_{d,h(d)} = 0$$

- Lemma: The dimension of the solution space of above system is equal to $d - 1$.

# 3. Equivalence test for NW

- Recall,

# 3. Equivalence test for NW

- Recall,

Black box of $g \in \mathbb{F}[\mathbf{x}]$ → Algorithm $\mathbb{A}$: Is $g$ equivalent to $f$ ?

Yes → $B$, s.t. $g = f(B \cdot \mathbf{x})$

No → 0 otherwise.

- Lemma:

If $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an arbitrary invertible matrix

Randomized polynomial time reduction ⇒

$f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible 'block permuted matrix'

# 3. Equivalence test for NW

▷ Recall,

Black box of $g \in \mathbb{F}[\mathbf{x}]$ → Algorithm $\mathcal{A}$: Is $g$ equivalent to $f$?

Yes → $B$, s.t. $g = f(B \cdot \mathbf{x})$

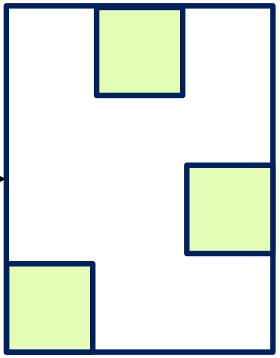
No → 0 otherwise.

▷ **Lemma:**

If $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an arbitrary invertible matrix

Randomized polynomial time reduction ⇒

$f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible 'block permuted matrix'



# 3. Equivalence test for NW

- Recall,

Black box of $g \in \mathbb{F}[\mathbf{x}]$ → Algorithm $\mathcal{A}$: Is $g$ equivalent to $f$ ?

Yes → $B$, s.t. $g = f(B \cdot \mathbf{x})$

No → 0 otherwise.

- Lemma:

If $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an arbitrary invertible matrix

$\xrightarrow{\text{Randomized polynomial time reduction}}$

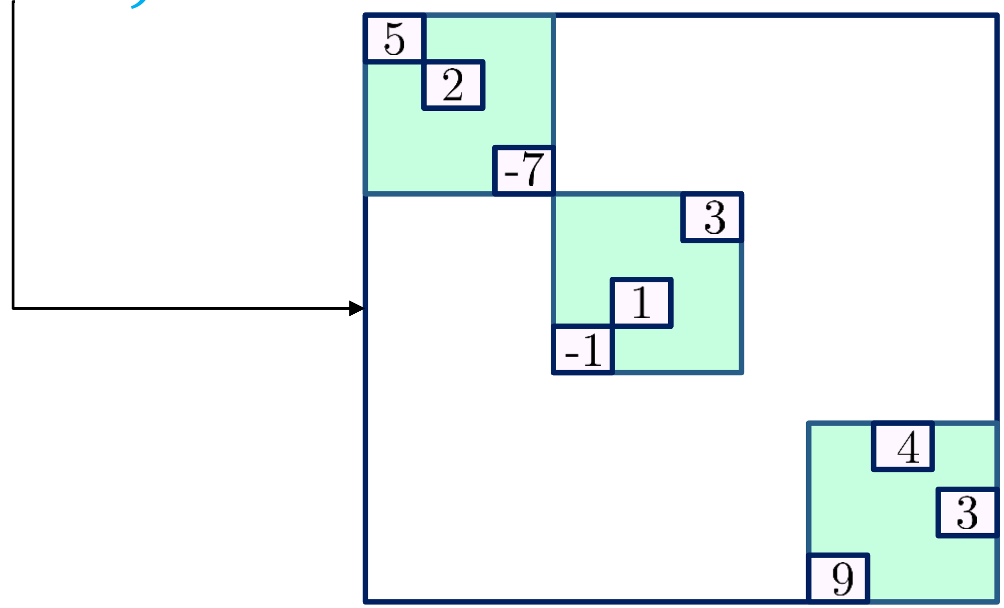
$f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible 'block permuted matrix'

- Proved using the Lie algebra of NW.

# 3. Equivalence test for NW

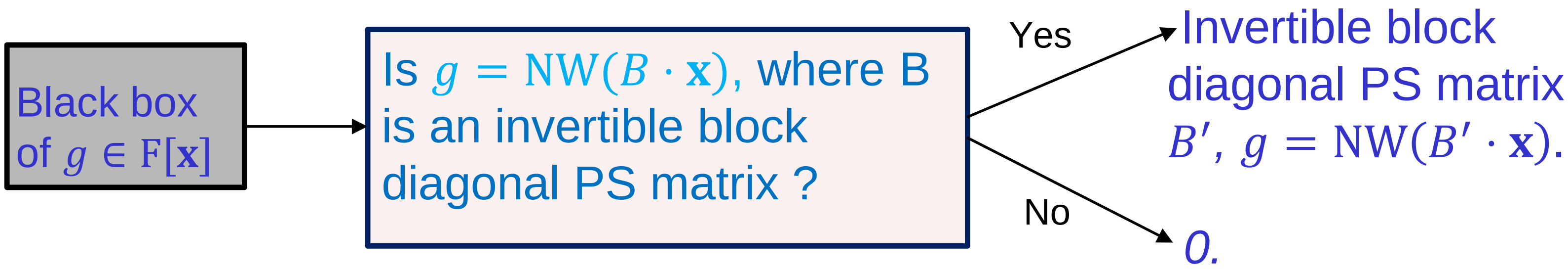
- We solve a special case of the equivalence test, called the block diagonal PS equivalence test, which is as follows:

- Is $f = \mathrm{NW}(B \cdot \mathbf{x})$?



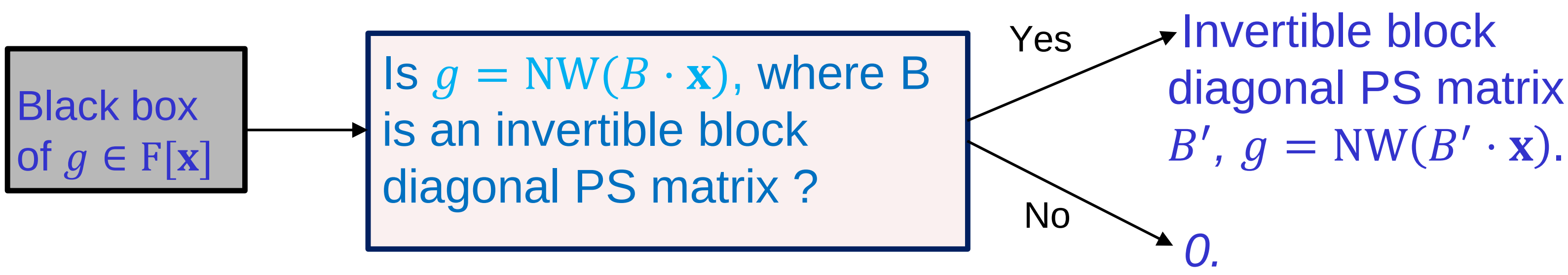
# 3.a PS Equivalence test for NW

- **Theorem**: Let F=**R** or finite field. Then, there exists a randomized polynomial time algorithm, such that



# 3.a PS Equivalence test for NW

- **Theorem**: Let F=R or finite field. Then, there exists a randomized polynomial time algorithm, such that




- We show the proof in 2 steps:
  - Solving 'Scaling equivalence test'.
  - Solving 'block diagonal permutation equivalence test'.

- Knowledge of symmetries of NW help in this.

# 3.b Scaling Eq. test for NW over $\mathbf{R}$

- Goal: To design an algorithm that determines if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible diagonal matrix.

# 3.b Scaling Eq. test for NW over $\mathbf{R}$

- Goal: To design an algorithm that determines if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible diagonal matrix.

Theorem [BRS17]: If $f, g \in \mathbf{R}[x]$ given as black boxes, then there exists a randomized polynomial time algorithm that checks if $f$ is scaling equivalent to $g$.

*Testing polynomial equivalence by scaling matrices* by Bläser, Rao and Sarma.

# 3.b Scaling Eq. test for NW over **R**

- Goal: To design an algorithm that determines if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible diagonal matrix.

  Theorem [BRS17]: If $f, g \in \mathbf{R}[\boldsymbol{x}]$ given as black boxes, then there exists a randomized polynomial time algorithm that checks if $f$ is scaling equivalent to $g$.

- It is not clear how to use this here as we do not have black box access to NW.

# 3.b Scaling Eq. test for NW over $\mathbf{R}$

- **Goal**: To design an algorithm that determines if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible diagonal matrix.

- **Idea** : Let $B = \mathrm{diag}\left((-1)^{s_{1,1}} \cdot 2^{b_{1,1}}, \ldots, (-1)^{s_{d,d}} \cdot 2^{b_{d,d}}\right) \in \mathbf{R}^{d^2 \times d^2}$ .

$$x_{1,h(1)} \cdots x_{d,h(d)} \xRightarrow{B \cdot \mathbf{x}} (-1)^{\lambda_h} \cdot 2^{\alpha_h} \cdot x_{1,h(1)} \cdots x_{d,h(d)}$$

# 3.b Scaling Eq. test for NW over $\mathbf{R}$

- Goal: To design an algorithm that determines if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible diagonal matrix.

- Idea : Let $B = \mathrm{diag}\left((-1)^{s_{1,1}} \cdot 2^{b_{1,1}}, \dots, (-1)^{s_{d,d}} \cdot 2^{b_{d,d}}\right) \in \mathbf{R}^{d^2 \times d^2}$.

$$x_{1,h(1)} \cdots x_{d,h(d)} \xrightarrow{B \cdot \mathbf{x}} (-1)^{\lambda_h} \cdot 2^{\alpha_h} \cdot x_{1,h(1)} \cdots x_{d,h(d)}$$

This implies,

$$b_{1,h(1)} + \cdots + b_{d,h(d)} = \alpha_h \text{ and}$$
$$s_{1,h(1)} + \cdots + s_{d,h(d)} = \lambda_h .$$

# 3.b Scaling Eq. test for NW over **R**

- **Goal**: To design an algorithm that determines if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is an invertible diagonal matrix.

- **Idea**: Let $B = \mathrm{diag}\left((-1)^{s_{1,1}} \cdot 2^{b_{1,1}}, \ldots, (-1)^{s_{d,d}} \cdot 2^{b_{d,d}}\right) \in \mathbf{R}^{d^2 \times d^2}$.

$$x_{1,h(1)} \cdots x_{d,h(d)} \xRightarrow{B \cdot \mathbf{x}} (-1)^{\lambda_h} \cdot 2^{\alpha_h} \cdot x_{1,h(1)} \cdots x_{d,h(d)}$$

Solved using the lemma used in the analysis of $g_{NW}$

This implies,

$$b_{1,h(1)} + \cdots + b_{d,h(d)} = \alpha_h \text{ and}$$

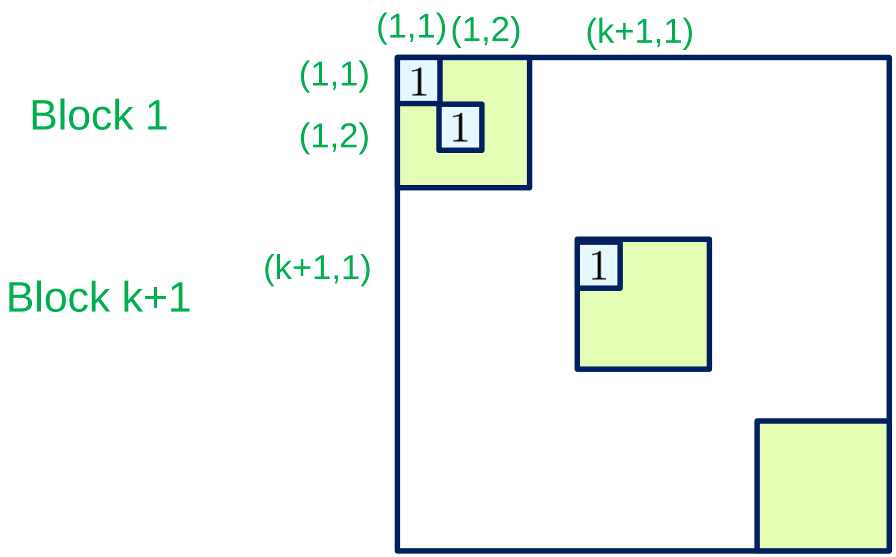
$$s_{1,h(1)} + \cdots + s_{d,h(d)} = \lambda_h .$$

# 3.c Block diagonal Permutation Eq. test

- Goal: To design a randomized poly time algorithm to determine if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is a block diagonal permutation matrix.

# 3.c Block diagonal Permutation Eq. test

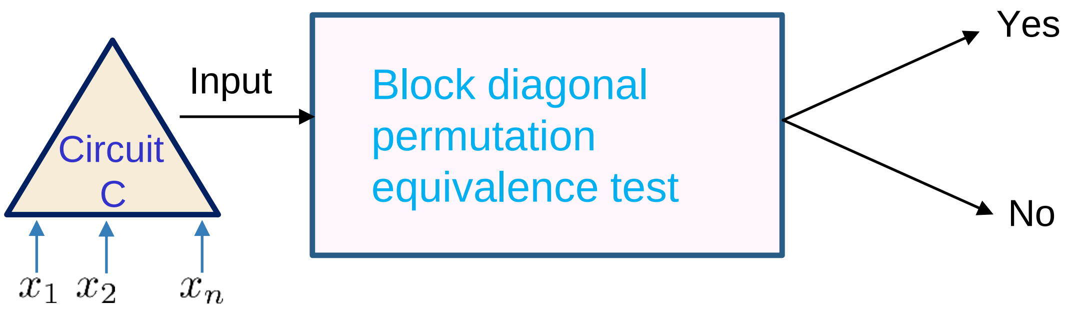
- **Goal**: To design a randomized poly time algorithm to determine if $f = \mathrm{NW}(B \cdot \mathbf{x})$, where $B$ is a block diagonal permutation matrix.
- **Proof Idea**: We can assume that $B$ looks as follows:




- In the first k+1 blocks on the diagonal of B, some exactly k+2 entries are assumed to be 1.
- No assumption is made on the other blocks.

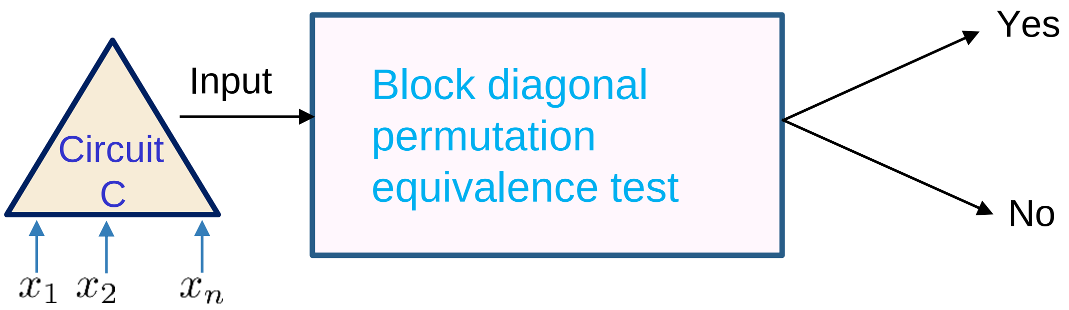
# 3.c Block diagonal Permutation Eq. test

- The same algorithm can also be used for circuit testing of NW.



# 3.c Block diagonal Permutation Eq. test

- The same algorithm can also be used for circuit testing of NW.




- We also have another algorithm for the circuit testing of NW, that uses the fact that 'NW is characterized by its circuit identities'.

# Open Questions

- Knowledge of all the permutation symmetries of NW.
- Equivalence test of NW:
  - Block diagonal equivalence test.
- Zero testing of NW.
- Circuit complexity of NW.
- Circuit compression for NW.

# Zero Testing of NW on $\{0,1\}^{d^2}$

- Recall,

$$\boldsymbol{a} \in \{0,1\}^{d^2} \longrightarrow \boxed{\text{Algorithm } \mathbb{A} \text{ for } \mathrm{NW}} \longrightarrow \begin{cases} 1, & \mathrm{NW}(\boldsymbol{a}) \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

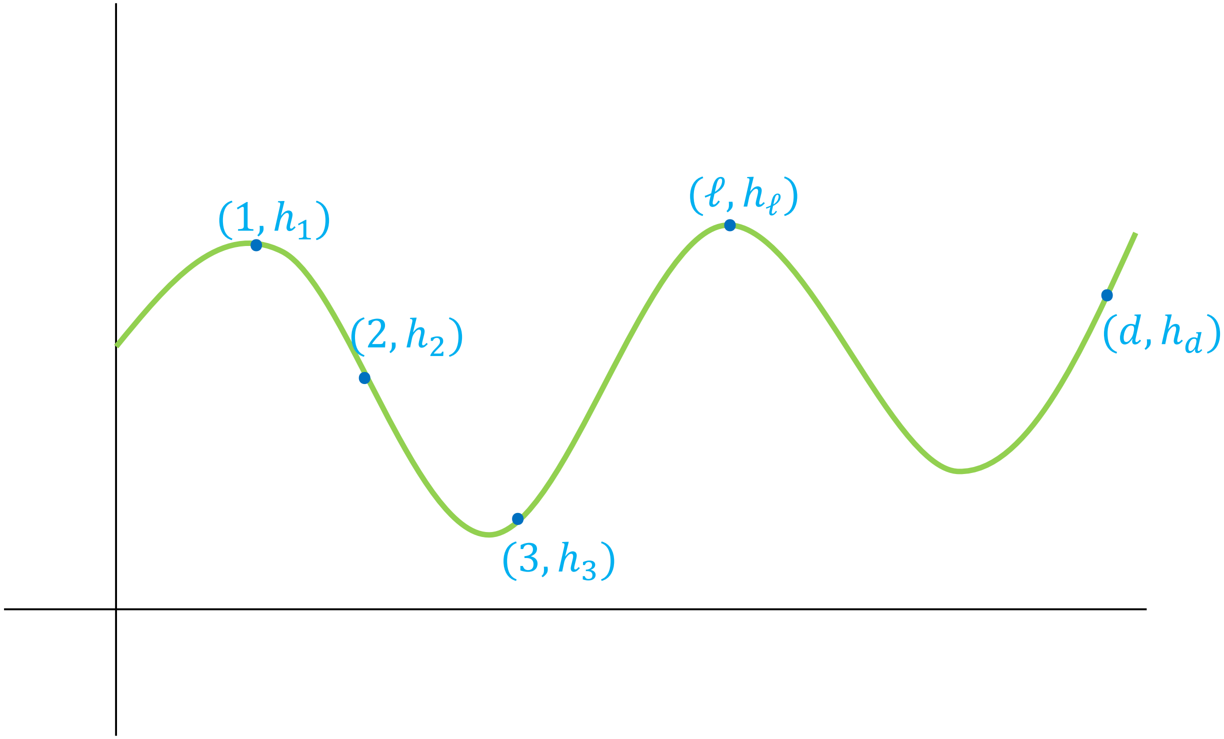
$$(\underbrace{0,1\cdots,1}_{\boldsymbol{a}_1},\underbrace{1,\cdots,0}_{\boldsymbol{a}_2},\cdots,\underbrace{1,\cdots,0}_{\boldsymbol{a}_d})$$

$$\boldsymbol{a}_1 \to S_1 \qquad \boldsymbol{a}_2 \to S_2 \qquad \boldsymbol{a}_d \to S_d$$

- Each $S_i \subseteq [d]$ , such that $\forall j \in S_i$, $\boldsymbol{a}_i(j) = 1$.

# Zero Testing of NW on $\{0,1\}^{d^2}$

- **Coding theoretic view:**
  Given $S_i \subseteq [d], i \in [d]$, does there exist an $h \in \mathrm{F}_d[z]_k$, such that $\forall\, \ell \in \mathrm{F}_d, h(\ell) \in S_\ell$?



# Thanks!

## Any questions?