Derandomizing PIT: A Survey of Results and Techniques

Bhargav Thankey
Indian Institute of Science, Bengaluru
Outline

• The PIT problem

• PIT and circuit lower bounds

• PIT for constant depth circuits

• PIT for constant read circuits

• PIT for orbits of circuit classes
Polynomial Identity Testing (PIT)

• The Problem: Given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, check if $f$ is identically zero.
• **The Problem:** Given a polynomial \( f \in \mathbb{F}[x_1, \ldots, x_n] \), check if \( f \) is identically zero.

The coefficients of all monomials are 0. Denoted \( f \equiv 0 \).

Not the same as \( f(a_1, \ldots, a_n) = 0 \) \( \forall a_1, \ldots, a_n \in \mathbb{F} \). Eg. \( x^2 - x \) over \( \mathbb{F}_2 \).
Polynomial Identity Testing (PIT)

• The Problem: Given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, check if $f$ is identically zero.

List of coefficients: Problem trivial
Polynomial Identity Testing (PIT)

• **The Problem:** Given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, check if $f$ is identically zero.
Polynomial Identity Testing (PIT)

• **The Problem:** Given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, check if $f$ is identically zero.

Arithmetic circuit

Black box access

\[
f(a_1, \ldots, a_n)
\]
Polynomial Identity Testing (PIT)

• The Problem: Given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, check if $f$ is identically zero.

White box PIT

Black box PIT

along with $n, d, s$. 
Polynomial Identity Testing (PIT)

- **The Problem**: Given a polynomial \( f \in \mathbb{F}[x_1, \ldots, x_n] \), check if \( f \) is identically zero.

White box PIT

Hitting sets

\[ f(a_1, \ldots, a_n) \]

along with \( n, d, s \).
Efficient randomised algorithm

• **Schwartz-Zippel Lemma** [DL78, Zip79, Sch80]: Let \( f \in \mathbb{F}[x_1, \ldots, x_n] \) be a non-zero, degree \( d \) polynomial. Then, for any \( S \subseteq \mathbb{F} \) and \( a_1, \ldots, a_n \in R \ S \),
  \[
  \Pr[f(a_1, \ldots, a_n) \neq 0] \geq 1 - \frac{d}{|S|}.
  \]

• Gives a \( \text{poly}(n, d) \) randomised algorithm for PIT: Pick \( a_1, \ldots, a_n \) uniformly at random from a large enough subset of \( \mathbb{F} \) and check whether \( f(a_1, \ldots, a_n) \) is 0.

• **Goal**: Obtain an efficient, deterministic algorithm for PIT.
Efficient randomised algorithm

- **Schwartz-Zippel Lemma** [DL78, Zip79, Sch80]: Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero, degree $d$ polynomial. Then, for any $S \subseteq \mathbb{F}$ and $a_1, \ldots, a_n \in_R S$,
  \[
  \Pr[f(a_1, \ldots, a_n) \neq 0] \geq 1 - \frac{d}{|S|}.
  \]

- Gives a $\text{poly}(n,d)$ randomised algorithm for PIT: Pick $a_1, \ldots, a_n$ uniformly at random from a large enough subset of $\mathbb{F}$ and check whether $f(a_1, \ldots, a_n)$ is 0.

- **Goal**: Obtain an **efficient, deterministic** algorithm for PIT.

  Running time $= \text{poly}(n,d,s)$. 
Connections to other problems

- **Primality testing**: The AKS primality test was obtained by derandomizing an instance of PIT over a ring.

- **Perfect matchings**: The best known randomised parallel algorithm for finding perfect matchings in graphs uses PIT [MVV87]. Derandomizing PIT will give a deterministic parallel algorithm to find perfect matchings in graphs.

- **Polynomial factoring**: A deterministic algorithm for PIT would yield a deterministic algorithm for polynomial factorisation [KSS15].

---

MVV87: Mulmuley-Vazirani-Vazirani, STOC, 87.
KSS15: Kopparty-Saraf-Shpilka, CCC, 14.
PIT and circuit lower bounds

• **Theorem [KI03]:** If there is a sub-exponential time algorithm for PIT, then either:

  1. There is a function in **NEXP** that can not be computed by polynomial sized Boolean circuits or
  2. the permanent polynomial can not be computed by polynomial sized arithmetic circuits.

**KI03:** Kabanets-Impagliazzo, STOC, 03.
PIT and circuit lower bounds

• **Theorem [KI03]**: If there is a sub-exponential time algorithm for PIT, then either:

1. There is a function in **NEXP** that cannot be computed by polynomial sized Boolean circuits or
2. The permanent polynomial can not be computed by polynomial sized arithmetic circuits.

\[
\text{Perm} \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,n} \end{bmatrix} := \sum_{\sigma \in S_n} \prod_{i \in [n]} x_{i,\sigma(i)}
\]
PIT and circuit lower bounds

• **Theorem** [KI03]: If there is a sub-exponential time algorithm for PIT, then either:

  1. There is a function in **NEXP** that cannot be computed by polynomial sized Boolean circuits or
  2. the permanent polynomial cannot be computed by polynomial sized arithmetic circuits.

• The result applies to both the white box and the black box setting.
PIT and circuit lower bounds

- **Theorem [HS80, Agr05]**: Let $T : \mathbb{N} \to \mathbb{N}$ be an increasing function. Suppose there is an algorithm which runs in time $T(s)$ and solves the black box version of PIT for size $s$ circuits. Then there exists an $n$ variate polynomial whose coefficients can be computed in time $2^{O(n)}$ that requires arithmetic circuits of size at least $T^{-1}(2^{O(n)})$.

HS80: Heintz-Schnorr, STOC, 80.
Agr05: Agrawal, FSTTCS, 05.
PIT and circuit lower bounds

• **Theorem** [HS80, Agr05]: Let $T : \mathbb{N} \rightarrow \mathbb{N}$ be an increasing function. Suppose there is an algorithm which runs in time $T(s)$ and solves the black box version of PIT for size $s$ circuits. Then there exists an $n$ variate polynomial whose coefficients can be computed in time $2^{O(n)}$ that requires arithmetic circuits of size at least $T^{-1}(2^{O(n)})$.

• **Polynomial time** black box PIT $\implies$ **exponential** arithmetic circuit lower bound.

• **Quasi-polynomial time** black box PIT $\implies$ arithmetic circuit lower bound of the form $2^{n^\epsilon}$.
PIT and circuit lower bounds

• **Theorem [KI03]:** If there is an $n$ variate, multilinear polynomial that requires arithmetic circuits of size $2^{\Omega(n)}$ (resp. $n^{\omega(1)}$), then there is a $2^{\text{polylog}(n)}$ (resp. sub-exponential) time black box PIT algorithm for $\text{poly}(n)$ sized arithmetic circuits computing $n$ variate polynomials of $\text{poly}(n)$ degree.

• Thus, derandomizing PIT and proving arithmetic circuit lower bounds are two sides of the same coin.
PIT for special circuit classes

• Since proving arithmetic circuit lower bounds seems to be difficult, we can expect derandomizing PIT to be a challenging problem.

• So the focus has been on derandomizing PIT for special classes of circuits.

• Some restrictions that have been imposed are:
  • Restricting the depth of the circuit,
  • Restricting the number of times the circuit can read a variable,
  • Restricting the fan-in of the gates in the circuit,
  • Combinations of the above three, etc.
PIT for Constant Depth Circuits
Constant depth circuits

- Alternating layers/levels of $+$ and $\times$ gates with unbounded fan-in.

$$\{x_1, \ldots, x_n\} \cup \mathbb{F}$$
Constant depth circuits

- Alternating layers/levels of $+$ and $\times$ gates with unbounded fan-in.

- Every layer of $+$ gates is denoted by $\Sigma$. Every layer of $\times$ gates is denoted by $\Pi$.

- Every depth $\Delta$ circuit can be denoted by a string of length $\Delta$ consisting of alternating $\Sigma$s and $\Pi$s.

\[ \{x_1, \ldots, x_n\} \cup \mathbb{F} \]
Constant depth circuits

- Alternating layers/levels of $+$ and $\times$ gates with unbounded fan-in.

- Every layer of $+$ gates is denoted by $\Sigma$. Every layer of $\times$ gates is denoted by $\Pi$.

- Every depth $\Delta$ circuit can be denoted by a string of length $\Delta$ consisting of alternating $\Sigma$s and $\Pi$s.

A $\Sigma\Pi\Sigma$ circuit
ΣΠ circuits

- A ΣΠ circuit (aka a sparse polynomial) computes an \( \mathbb{F} \)-linear combination of monomials and is thus a universal model of computation.

- White box PIT: Trivial.

- Black box PIT [KS01]: There is a poly\((n, d, s)\) time black box PIT algorithm for the class of \(n\) variate, degree \(d\), \(s\) sparse polynomials over fields of size poly\((n, d, s)\).

KS01: Klivans-Spielman, STOC, 01.
Let $f = \sum_{i \in [s]} c_i \cdot x_1^{d_{i,1}} \cdots x_n^{d_{i,n}}$ be a non-zero, degree $d$, $s$ sparse polynomial.

Map $x_i \mapsto x^{t^{i-1} \mod q}, \forall i \in [n]$, where $q$ is a prime number $> s^2 nd$. Thus the monomial $x_1^{d_{i,1}} \cdots x_n^{d_{i,n}}$ maps to $x^{d_{i,n}(t^{n-1} \mod q) + d_{i,n-1}(t^{n-2} \mod q) + \cdots + d_{i,1}}$.

Let $p_i(t) = d_{i,n}(t^{n-1} \mod q) + d_{i,n-1}(t^{n-2} \mod q) + \cdots + d_{i,1}$. We find an $\alpha \in \mathbb{N}$ s.t. $\forall i \neq j, p_i(\alpha) \neq p_j(\alpha) \mod q$. Then $\forall i \neq j, p_i(\alpha) \neq p_j(\alpha)$. 
Let \( f = \sum_{i \in [s]} c_i \cdot x_1^{d_{i,1}} \cdots x_n^{d_{i,n}} \) be a non-zero, degree \( d, s \) sparse polynomial.

Map \( x_i \mapsto x^{t_{i-1} \mod q}, \forall i \in [n] \), where \( q \) is a prime number > \( s^2 nd \). Thus the monomial \( x_1^{d_{i,1}} \cdots x_n^{d_{i,n}} \) maps to \( x^{d_{i,n}(t^{n-1} \mod q)+d_{i,n-1}(t^{n-2} \mod q)+\cdots+d_{i,1}} \).

Let \( p_i(t) = d_{i,n}(t^{n-1} \mod q)+d_{i,n-1}(t^{n-2} \mod q)+\cdots+d_{i,1} \). We find an \( \alpha \in \mathbb{N} \) s.t. \( \forall i \neq j, p_i(\alpha) \neq p_j(\alpha) \mod q \). Then \( \forall i \neq j, p_i(\alpha) \neq p_j(\alpha) \).

Any \( \alpha \) which is not a root of \( \prod_{i \neq j} (p_i(t) - p_j(t)) \mod q \) over \( \mathbb{F}_q \) will work. As \( q > s^2 n \) such an \( \alpha \) exists.
ΣΠ circuits – black box PIT

• Let \( f = \sum_{i \in [s]} c_i \cdot x_1^{d_i,1} \cdots x_n^{d_i,n} \) be a non-zero, degree \( d, s \) sparse polynomial.

• Map \( x_i \mapsto x^{t_{i}^{-1} \mod q}, \forall i \in [n] \), where \( q \) is a prime number \( > s^2nd \). Thus the monomial \( x_1^{d_i,1} \cdots x_n^{d_i,n} \) maps to \( x^{d_i,n(t^{n-1} \mod q)+d_{i,n-1}(t^{n-2} \mod q)+ \cdots +d_{i,1}} \).

• Let \( p_i(t) = d_{i,n}(t^{n-1} \mod q)+d_{i,n-1}(t^{n-2} \mod q)+ \cdots +d_{i,1} \). We find an \( \alpha \in \mathbb{N} \) s.t. \( \forall i \neq j, p_i(\alpha) \neq p_j(\alpha) \mod q \). Then \( \forall i \neq j, p_i(\alpha) \neq p_j(\alpha) \).

• Now \( f(x, x^{\alpha \mod q}, \ldots, x^{\alpha^{n-1} \mod q}) \) is a non-zero, univariate polynomial of degree \( \leq dq \). Thus, by trying out \( \leq dq + 1 \) many values for \( x \), we find a \( \beta \in \mathbb{F} \) s.t. \( f(\beta, \beta^{\alpha \mod q}, \ldots, \beta^{\alpha^{n-1} \mod q}) \neq 0 \).
Let $f = \sum_{i \in [s]} c_i \cdot x_1^{d_{i,1}} \cdots x_n^{d_{i,n}}$ be a non-zero, degree $d$, $s$ sparse polynomial.

Map $x_i \mapsto x^{t^{i-1} \mod q}, \forall i \in [n]$, where $q$ is a prime number $> s^2nd$. Thus the monomial $x_1^{d_{i,1}} \cdots x_n^{d_{i,n}}$ maps to $x^{d_{i,n}(t^{n-1} \mod q)+d_{i,n-1}(t^{n-2} \mod q)+\cdots+d_{i,1}}$.

Let $p_i(t) = d_{i,n}(t^{n-1} \mod q)+d_{i,n-1}(t^{n-2} \mod q)+\cdots+d_{i,1}$. We find an $\alpha \in \mathbb{N}$ s.t. $\forall i \neq j, p_i(\alpha) \neq p_j(\alpha) \mod q$. Then $\forall i \neq j, p_i(\alpha) \neq p_j(\alpha)$.

Now $f(x, x^\alpha \mod q, \ldots, x^{\alpha^{n-1} \mod q})$ is a non-zero, univariate polynomial of degree $\leq dq$. Thus, by trying out $\leq dq + 1$ many values for $x$, we find a $\beta \in \mathbb{F}$ s.t. $f(\beta, \beta^\alpha \mod q, \ldots, \beta^{\alpha^{n-1} \mod q}) \neq 0$.

Such a $\beta$ will exist as $|\mathbb{F}| = \text{poly}(n, d, s)$. 
• **Running time of the algorithm:** The algorithm finds \( q \), tries at most \( s^2n + 1 \) many values of \( \alpha \) and for each value of \( \alpha \), tries at most \( dq + 1 \) many values of \( \beta \).

• A prime \( s^2nd < q \leq 2s^2nd \) exists and can be found in \( \text{poly}(n, d, s) \) time. Time required to try various values of \( \alpha \) and \( \beta \) is \( \leq (s^2n + 1)(dq + 1) = \text{poly}(n, d, s) \). Total time = \( \text{poly}(n, d, s) \).
**ΣΠΣ circuits**

- **Theorem** [VSBR83, AV08, Koi12, GKKS13, Tav13]: If $f$ is an $n$ variate, degree $\text{poly}(n)$ polynomial computed by a $\text{poly}(n)$ size circuit, then it can also be computed by a $\Sigma\Pi\Sigma$ circuit of size $n^{O(\sqrt{n})}$.

- **Polynomial time PIT** for $\Sigma\Pi\Sigma$ circuits $\Rightarrow$ **sub-exponential PIT** for $\text{poly}(n)$ size circuits computing $\text{poly}(n)$ degree polynomials. PIT for $\Sigma\Pi\Sigma$ circuits is as challenging as PIT for general circuits.

- Researchers have studied restricted classes of $\Sigma\Pi\Sigma$ circuits.

VSBR83: Valiant-Skyum-Berkowitz-Rackoff, SICOMP, 83.
AV08: Agrawal-Vinay, FOCS, 08.
\( \Sigma^k \Pi^d \Sigma \) circuits

- A \( \Sigma^k \Pi^d \Sigma \) circuit is a \( \Sigma \Pi \Sigma \) circuit where the fan-in of the top + gate is at most \( k \) and the fan-in of all product gates in the second level is at most \( d \). Think of \( k \) as a constant.

- Both white box and black box PIT for \( \Sigma^k \Pi^d \Sigma \) circuits have been studied extensively.
PIT for $\Sigma^k \Pi^d \Sigma$ circuits

<table>
<thead>
<tr>
<th>Paper</th>
<th>Version</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS05</td>
<td>White box</td>
<td>$\text{poly}(n, d^{O(k^2 \log k - 2d)})$</td>
</tr>
<tr>
<td>KS06</td>
<td>White box</td>
<td>$\text{poly}(n, d^{O(k)})$</td>
</tr>
<tr>
<td>KS08</td>
<td>Black box</td>
<td>$\text{poly}(n, d^{O(k^2 \log k - 2d)})$</td>
</tr>
<tr>
<td>SS09</td>
<td>Black box</td>
<td>$\text{poly}(n, d^{O(k^3 \log d)})$</td>
</tr>
<tr>
<td>KS09</td>
<td>Black box</td>
<td>$\text{poly}(n, d^{O(k^k)})$ over $\mathbb{R}$</td>
</tr>
<tr>
<td>SS10</td>
<td>Black box</td>
<td>$\text{poly}(n, d^{O(k^2)})$ over $\mathbb{R}$, $\text{poly}(n, d^{O(k^2 \log d)})$ over any $\mathbb{F}$</td>
</tr>
<tr>
<td>SS11</td>
<td>Black box</td>
<td>$\text{poly}(n, d^{O(k)})$</td>
</tr>
</tbody>
</table>

DS05: Dvir-Shpilka, STOC, 05.
KS06: Kayal-Saxena, CCC, 06.
KS08: Karnin-Shpilka, CCC, 08.
SS09: Saxena-Seshadhri, CCC, 09.
KS09: Kayal-Saraf, FOCS, 09.
SS10: Saxena-Seshadhri, FOCS, 10.
SS11: Saxena-Seshadhri, STOC, 11.
An approach for \( \Sigma^k \Pi^d \Sigma \) black box PIT

- Let \( f = T_1 + \cdots + T_k, \ T_i = \ell_{i,1} \cdots \ell_{i,d_i}, \) where \( \ell_{i,j} \) are linear polynomials, be a \( \Sigma^k \Pi^d \Sigma \) circuit computing an \( n \) variate polynomial.
An approach for $\Sigma^k \Pi^d \Sigma$ black box PIT

• Let $f = T_1 + \cdots + T_k, T_i = \ell_{i,1} \cdots \ell_{i,d}$, where $\ell_{i,j}$ are linear forms, be a $\Sigma^k \Pi^d \Sigma$ circuit computing an $n$ variate polynomial.

• A lot of black box PIT algorithms for $\Sigma^k \Pi^d \Sigma$ circuits use the rank bound idea.

• $\text{rank}(f) := \text{dim span}\{\ell_{1,1}, \ldots, \ell_{k,d}\}$. 
rank and $\Sigma^k \Pi^d \Sigma \Pi T$

• Suppose $\text{rank}(f) = r$. Let $\{\ell_{i_1,j_1}, ..., \ell_{i_r,j_r}\}$ be a basis of $\text{span}\{\ell_{1,1}, ..., \ell_{k,d}\}$.

• **Rank extractors:** Let $V$ be an unknown but fixed space of linear functions from $\mathbb{F}^n$ to $\mathbb{F}$ of dimension at most $r$. [GR05] showed that a linear transformation $T: \mathbb{F}^r \rightarrow \mathbb{F}^n$ s.t. $\dim V \circ T = \dim V$ can be constructed in $\text{poly}(n, r)$ time provided that $|\mathbb{F}| = \text{poly}(n, r)$.

• $V := \text{span}\{\ell_{i_1,j_1}, ..., \ell_{i_r,j_r}\}$. It is not too difficult to show that $f \equiv 0 \iff f \circ T \equiv 0$.

• $f \circ T$ is an $r$ variate polynomial. If $r$ is “small” we can find a non-root of $f \circ T$ by brute force search.

GR05: Gabizon-Raz, FOCS, 05.
We can not expect the rank of an arbitrary $\Sigma^k \Pi^d \Sigma$ circuit to be small.

However, it turns out that a rank bound for simple and minimal $\Sigma^k \Pi^d \Sigma$ circuits computing the 0 polynomial suffices.
We can not expect the rank of an arbitrary $\Sigma^k \Pi^d \Sigma$ circuit to be small.

However, it turns out that a rank bound for simple and minimal $\Sigma^k \Pi^d \Sigma$ circuits computing the 0 polynomial suffices.

$$f = T_1 + \cdots + T_k$$ is simple if there is no linear form that divides all of $T_1, \ldots, T_k$. 
We can not expect the rank of an arbitrary $\Sigma^k \Pi^d \Sigma$ circuit to be small.

However, it turns out that a rank bound for simple and minimal $\Sigma^k \Pi^d \Sigma$ circuits computing the 0 polynomial suffices.

$$f = T_1 + \cdots + T_k$$ is minimal if $\forall S \subseteq [k], \sum_{i \in S} T_i \neq 0.$
rank and $\Sigma^k \Pi^d \Sigma$ PIT

• We can not expect the rank of an arbitrary $\Sigma^k \Pi^d \Sigma$ circuit to be small.

• However, it turns out that a rank bound for simple and minimal $\Sigma^k \Pi^d \Sigma$ circuits computing the 0 polynomial suffices.

• **Theorem** [KS06]: Suppose that the rank of all $n$ variate simple and minimal $\Sigma^k \Pi^d \Sigma$ circuits computing the 0 polynomial is at most $R(k, d)$. Then, there is an $\text{poly}(n, 2^k, d^{R(k,d)})$ time black box PIT algorithm for $\Sigma^k \Pi^d \Sigma$ circuits.

• The proof of the above theorem crucially uses the rank extractors from [GR05].

KS06: Karnin-Shpilka, CCC, 06.
GR05: Gabizon-Raz, FOCS, 05.
rank and $\Sigma^k \Pi^d \Sigma$ PIT

• How can we show that the rank of every simple and minimal $\Sigma^k \Pi^d \Sigma$ circuit computing the 0 polynomial is “small”?

• One way is to use Sylvester-Gallai type theorems.
A detour: Sylvester-Gallai theorem

- **Sylvester-Gallai Theorem**: Let $S \subseteq \mathbb{R}^2$ be a finite set. If $\forall a, b \in S, \exists c \in S$, s.t. the line passing through $a$ and $b$ also contains $c$, then all points in $S$ are collinear.

- **Edelstein-Kelly Theorem**: Let $R, G, B \subseteq \mathbb{R}^2$ be disjoint, finite sets of the same size. If for every pair of points $a, b$ from two distinct sets, there exists $c$ in the third set, s.t. the line passing through $a$ and $b$ also contains $c$, then all points in $R \cup G \cup B$ are collinear.
• How can we show that the rank of every simple and minimal $\Sigma^k \Pi^d \Sigma$ circuit computing the 0 polynomial is “small”?

• Let $f = T_1 + T_2 + T_3$ be a simple and minimal $\Sigma^k \Pi^d \Sigma$ circuit computing the 0 polynomial. Let $T_i = \ell_{i,1} \cdots \ell_{i,d}$ and $S_i = \{\ell_{i,1}, \ldots, \ell_{i,d}\}$. Since $f$ is simple, the $S_i$ are disjoint. Now, $0 \equiv f \mod \ell_{1,1} = (T_2+T_3) \mod \ell_{1,1} \implies \forall \ell_{2,j}, \exists \ell_{3,j'}$ s.t. $\ell_{3,j'} = \ell_{2,j} \mod \ell_{1,1}$. I.e. $\ell_{3,j'} \in \text{span}\{\ell_{2,j}, \ell_{1,1}\}$. Thus, $S_1, S_2, S_3$ have a structure like the one found in the hypothesis of the Edelstein-Kelly Theorem. Perhaps this can be used to bound the rank.

• Sylvester-Gallai type theorems were used to bound rank in [KS09, SS10].

KS09: Kayal-Saraf, FOCS, 09.
SS10: Saxena-Seshadhri, FOCS, 10.
\[ \Sigma^k \Pi^d \Sigma \text{ black box PIT} \]

- **Summary:**

1. **Rank bound** on simple, minimal $\Sigma^k \Pi^d \Sigma$ circuits computing the 0 polynomial + **Rank extractors** imply black box PIT for $\Sigma^k \Pi^d \Sigma$ circuits.

2. **Sylvester-Gallai** type theorems can be used to prove that the rank of simple, minimal $\Sigma^k \Pi^d \Sigma$ circuits computing the 0 polynomial is “small”.
\( \Sigma \land \Sigma \) circuits

- \( \Sigma \land \Sigma \) circuits are a natural sub-class of \( \Sigma \Pi \Sigma \) circuits.

- A \( \Sigma \land \Sigma \) circuit looks like \( \sum_{i \in [k]} \ell_i^d \). I.e. all the inputs of a \( \times \) gate in the second level are the same.

- [Sax08, FS13] showed that \( \Sigma \land \Sigma \) circuits are a sub-class of Read-once Oblivious Algebraic Branching Programs (ROABPs).

- This observation yields polynomial time white box and quasi-polynomial time black box PIT algorithms for this model.

Sax08: Saxena, ICALP, 08.
Depth 4 circuits

• **Theorem** [VSBR83, AV08, Koi12, GKKS13, Tav13]: If $f$ is an $n$ variate, degree $\text{poly}(n)$ polynomial computed by a $\text{poly}(n)$ size circuit, then it can also be computed by a $\Sigma\Pi\Sigma\Pi$ circuit of size $n^{O(\sqrt{n})}$.

---

AV08: Agrawal-Vinay, FOCS, 08.
Depth 4 circuits

- **Theorem** [VSBR83, AV08, Koi12, GKKS13, Tav13]: If $f$ is an $n$ variate, degree $\text{poly}(n)$ polynomial computed by a $\text{poly}(n)$ size circuit, then it can also be computed by a $\Sigma\Pi\Sigma\Pi$ circuit of size $n^{O(\sqrt{n})}$.

  In fact, by circuits where $\times$ gates have fan-in $O(\sqrt{n})$.

AV08: Agrawal-Vinay, FOCS, 08.
Depth 4 circuits

- **Theorem** [VSBR83, AV08, Koi12, GKKS13, Tav13]: If \( f \) is an \( n \) variate, degree \( \text{poly}(n) \) polynomial computed by a \( \text{poly}(n) \) size circuit, then it can also be computed by a \( \Sigma \Pi \Sigma \Pi \) circuit of size \( n^{O(\sqrt{n})} \).

- Polynomial time PIT for \( \Sigma \Pi \Sigma \Pi \) circuits \( \Rightarrow \) sub-exponential PIT for \( \text{poly}(n) \) size circuits computing \( \text{poly}(n) \) degree polynomials.

- A natural sub-class to study is \( \Sigma^k \Pi \Sigma \Pi^\delta \) circuits.
PIT for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits?

• One natural approach is to generalise the notion of rank, rank extractors, and Sylvester-Gallai type theorems used for $\Sigma^k \Pi^d \Sigma$ circuits to appropriate notions for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits. This was done in [BMS11, Gup14].

• [BMS11] replaces rank by transcedence degree.

BMS11: Beecken-Mittmann-Saxena, ICALP, 11.
Gup14: Gupta, ECCC, 14.
A detour: algebraic independence

- $f_1, ..., f_m \in \mathbb{F}[x_1, ..., x_n]$ are said to be algebraically independent if there does not exist any non-zero $P \in \mathbb{F}[y_1, ..., y_m]$ s.t. $P(f_1, ..., f_m) \equiv 0$.

- $\mathbb{F}[x_1, ..., x_n]$ forms a matroid under algebraic independence.

- **Transcendence degree:** For any $S \subseteq \mathbb{F}[x_1, ..., x_n]$, the transcendence degree of $S$, denoted by $\text{tr} - \text{deg}(S)$, is the size of the maximum cardinality set of algebraically independent polynomials in $S$. It can be shown that $\text{tr} - \text{deg}(S) \leq n$. 
PIT for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits?

- One natural approach is to generalise the notion of rank, rank extractors, and Sylvester-Gallai type theorems used for $\Sigma^k \Pi^d \Sigma$ circuits to appropriate notions for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits. This was done in [BMS11, Gup14].

- [BMS11] replaces rank by transcendence degree. Let $f = \Sigma_{i \in [k]} \Pi_{j \in [s]} f_{i,j}$ be a $\Sigma^k \Pi \Sigma \Pi^\delta$ circuit. Then,

$$\text{rank}(f) := \text{tr} - \deg \{f_{i,j}\}_{i,j}.$$
PIT for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits?

- [BMS11] replaces rank extractors by faithful homomorphisms.

BMS11: Beecken-Mittmann-Saxena, ICALP, 11.
PIT for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits?

• [BMS11] replaces rank extractors by faithful homomorphisms.

\[ \phi : \mathbb{F}[x_1, \ldots, x_n] \rightarrow \mathbb{F}[y_1, \ldots, y_m] \quad \text{s.t.} \]
\[ \forall p, q \in \mathbb{F}[x_1, \ldots, x_n], \]
\[ \phi(p + q) = \phi(p) + \phi(q) \quad \text{and} \]
\[ \phi(pq) = \phi(p)\phi(q). \]
PIT for $\Sigma^k \Pi^p \Pi^\delta$ circuits?

- [BMS11] replaces rank extractors by faithful homomorphisms.

\[ \phi: \mathbb{F}[x_1, \ldots, x_n] \rightarrow \mathbb{F}[y_1, \ldots, y_m] \] is said to be faithful to \( \{f_1, \ldots, f_s\} \) if

\[ \text{tr} - \deg\{f_1, \ldots, f_s\} = \text{tr} - \deg\{\phi(f_1), \ldots, \phi(f_s)\}. \]
PIT for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits?

• [BMS11] replaces rank extractors by faithful homomorphisms.

• **Theorem** [BMS11]: If the rank of every $n$ variate simple and minimal size $s$ $\Sigma^k \Pi \Sigma \Pi^\delta$ circuit computing the 0 polynomial is at most $r$, then there is a black box PIT algorithm for size-$s$ $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits that runs in time $\text{poly}(n, r, \delta, s)\delta^2 kr$. 
PIT for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits?

• To bound the rank of simple, minimal $\Sigma^k \Pi \Sigma \Pi^\delta$ circuit computing the 0 polynomial, [Gup14] proposed a Sylvester-Gallai type conjecture for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits.

• [Shp19, PS20, PS21] proved Gupta’s conjecture for $\Sigma^3 \Pi \Sigma \Pi^2$ circuits thereby obtaining a black box, $\text{poly}(n, d)$ PIT algorithm for $\Sigma^3 \Pi \Sigma \Pi^2$ circuits.

Gup14: Gupta, ECCC, 14.
Shp19: Shpilka, FOCS, 19.
## PIT for depth 4 circuits

<table>
<thead>
<tr>
<th>Model</th>
<th>Paper</th>
<th>Version</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma^k \Pi^\delta \Sigma \land$</td>
<td>Sax08</td>
<td>White box</td>
<td>$\text{poly}(n, k, s^{O(\delta)})$</td>
</tr>
<tr>
<td>Multilinear $\Sigma^k \Pi \Sigma \Pi$</td>
<td>SV11, ASSS12</td>
<td>Black box</td>
<td>$\text{poly}(n^{O(k^2)})$</td>
</tr>
<tr>
<td>$\Sigma^2 \Pi \Sigma \Pi^\delta$</td>
<td>BMS11</td>
<td>Black box</td>
<td>$\text{poly}(n, \delta, s)^{\delta^2}$</td>
</tr>
<tr>
<td>$\Sigma \land \Sigma \Pi^\delta$</td>
<td>For15</td>
<td>Black box</td>
<td>$s^{O(\delta \log s)}$</td>
</tr>
<tr>
<td>$\Sigma^3 \Pi \Sigma \Pi^2$</td>
<td>PS21</td>
<td>Black box</td>
<td>$\text{poly}(n, d)$</td>
</tr>
<tr>
<td>$\Sigma^k \Pi \Sigma \land$</td>
<td>DDS20</td>
<td>White box</td>
<td>$s^{O(k 7^k)}$</td>
</tr>
<tr>
<td>$\Sigma^k \Pi \Sigma \land$</td>
<td>DDS20</td>
<td>Black box</td>
<td>$s^{O(k \log \log s)}$</td>
</tr>
<tr>
<td>$\Sigma^k \Pi \Sigma \Pi^\delta$</td>
<td>DDS20</td>
<td>Black box</td>
<td>$s^{O(\delta^2 k \log s)}$</td>
</tr>
<tr>
<td>$\Sigma^{k+1} \Pi \Sigma \land$</td>
<td>DDS21</td>
<td>Black box</td>
<td>$s^{O(k 7^k \log \log s)}$</td>
</tr>
<tr>
<td>$\Sigma^k \Pi \Sigma \Pi^\delta$</td>
<td>DDS21</td>
<td>Black box</td>
<td>$s^{O(\delta^2 k 7^k \log s)}$</td>
</tr>
</tbody>
</table>

Sax08: Saxena, ICALP, 08.
SV11: Saraf-Volkovich, STOC, 11.
ASSS12: Agrawal-Saha-Sapthirishy-Saxena, STOC, 12.
BMS11: Beecken-Mittmann-Saxena, ICALP, 11.
For15: Forbes, FOCS, 15.
PIT for low depth circuits

• In a breakthrough paper [LST21], Limaye, Srinivasan, and Tavenas proved super-polynomial lower bounds for low depth circuits.

• [DSY08, CKS19] showed that super-polynomial lower bounds for low depth circuits imply sub-exponential PIT for such circuits.

• Thus, [LST21] yields a \((n \cdot s^{\Delta + 1})^{n \epsilon}, \epsilon > 0\), time PIT for depth \(\Delta = o(\log \log \log n)\) circuits provided that \(s = \text{poly}(n)\).
PIT for Constant Read Circuits
Read once formulas

• **Arithmetic Formulas**: Arithmetic circuits whose underlying graph is a tree.
Read once formulas

• **Read Once Formulas (ROFs):** Arithmetic formulas where each variable appears in at most one leaf.

![Diagram of ROF example](image)

**ROF:**

- \( x_1 + x_2 \times x_3 \times x_5 + x_4 \times 2 \)

**Not an ROF:**

- \( x_1 + x_2 \times x_3 \times x_4 + x_3 \times x_4 + 3 \times x_5 \)
Read once formulas

• **Read Once Formulas (ROFs):** Arithmetic formulas where each variable appears in at most one leaf.

• ROFs are a special class of multilinear circuits.

• [SV09] gave an $n^{O(\log n)}$ time black box PIT algorithm for ROFs.

• This was improved to a $\text{poly}(n)$ time algorithm by [MV17].

SV09: Shpilka-Volkovich, APPROX-RANDOM, 09.
MV17: Minahan-Volkovich, CCC, 17.
Constant read formulas

• **Read $k$ Formulas**: Arithmetic formulas where each variable appears in at most $k$ leaves.

• [SV09] gave an $n^{O(k + \log n)}$ time black box PIT algorithm for sum of $k \leq \frac{n}{3}$ ROFs.

SV09: Shpilka-Volkovich, APPROX-RANDOM, 09.
Constant read formulas

• **Read $k$ Formulas**: Arithmetic formulas where each variable appears in at most $k$ leaves.

• [AvMV11] gave a $\text{poly}(s, n^{k^{O(k)}})$ time white box and $n^{k^{O(k)} + O(k \log n)}$ time black box PIT algorithm for multilinear read $k$ formulas.

AvMV11: Anderson-van Melkebeek-Volkovich, CCC, 11.
Constant read formulas

- **Read $k$ Formulas**: Arithmetic formulas where each variable appears in at most $k$ leaves.

- [ASSS12] gave an $s^{k^O(\Delta 2^\Delta)}$ time black box PIT algorithm for occur $k$ formulas of depth $\Delta$ using the algebraic independence technique from [BMS11].

ASSS12: Agrawal-Saha-Sapthirishi-Saxena, STOC, 12.
BMS11: Beecken-Mittmann-Saxena, ICALP, 11.
Constant read formulas

• **Read $k$ Formulas**: Arithmetic formulas where each variable appears in at most $k$ leaves.

• [ASSS12] gave an $s^{kO(\Delta^2)}$ time black box PIT algorithm for occur $k$ formulas of depth $\Delta$ using the algebraic independence technique from [BMS11].

A generalisation of read $k$ formulas. Capture other interesting models like multilinear $\Sigma^k \Pi \Sigma \Pi$ circuits.
Read-once oblivious algebraic branching programs

- **ROABP:** \( f \in \mathbb{F}[x_1, \ldots, x_n] \) is said to be computed by a width-\( w \) ROABP in order \( \pi \in S_n \) if

\[
f = [1, \ldots, 1]\left[ \begin{array}{c} M_1(x_{\pi(1)}) \cr \vdots \cr M_n(x_{\pi(n)}) \end{array} \right]_{w \times w} [1].
\]

- The classes of \( \Sigma \land \Sigma \) and \( \Sigma \land \Sigma \land \Sigma \) circuits are contained in ROABPs.

- [OSV15] obtained a sub-exponential time black box PIT for multilinear depth 3 and depth 4 formulas by reducing to black box PIT for ROABPs.

OSV15: Oliveira-Shpilka-Volk, CCC, 15.
Read-once oblivious algebraic branching programs

• A $\text{poly}(n, d, w)$ white box PIT for ROABPs follows from [RS04].
• [FS13] gave a $\text{poly}(n, d, w)^{O(\log w)}$ time black box PIT for ROABPs with known variable order.
• [FSS14] gave a $\text{poly}(n, d)^{O(\log w)}$ time black box PIT for multilinear and commutative ROABPs.

RS05: Raz-Shpilka, CCC, 04.
Read-once oblivious algebraic branching programs

- A $\text{poly}(n, d, w)$ white box PIT for ROABPs follows from [RS04].
- [FS13] gave a $\text{poly}(n, d, w)^{O(\log w)}$ time black box PIT for ROABPs with known variable order.
- [FSS14] gave a $\text{poly}(n, d)^{O(\log w)}$ time black box PIT for multilinear and commutative ROABPs.

$f$ is computed by a width $w$ commutative ROABP if it is computed by a width $w$ ROABP in every variable order.

RS05: Raz-Shpilka, CCC, 04.
Read-once oblivious algebraic branching programs

- A $\text{poly}(n, d, w)$ white box PIT for ROABPs follows from [RS04].
- [FS13] gave a $\text{poly}(n, d, w)^O(\log w)$ time black box PIT for ROABPs with known variable order.
- [FSS14] gave a $\text{poly}(n, d)^O(\log w)$ time black box PIT for multilinear and commutative ROABPs.
- [AGKS15] gave a $\text{poly}(n, d, w)^O(\log n)$ time black box PIT for ROABPs with unknown variable order.
- [GKST15] gave a $\text{poly}(n, d, w)^O(\log n)$ time black box PIT and $\text{poly}(n, d, w)$ time white box PIT for sum of constantly many ROABPs.

RS05: Raz-Shpilka, CCC, 04.
GKST15: Agrawal-Gurjar-Saxena-Thierauf, CCC, 15.
PIT for Orbits of Circuit Classes
Orbits

- **Orbit of a polynomial:** For $f \in \mathbb{F}[x_1, \ldots, x_n]$, the orbit of $f$, denoted by $\text{orb}(f)$, is the set $\{f(Ax + b) : A \in \text{GL}(n, \mathbb{F}) \text{ and } b \in \mathbb{F}^n\}$. 
Orbits

• **Orbit of a polynomial:** For $f \in \mathbb{F}[x_1, \ldots, x_n]$, the orbit of $f$, denoted by $\text{orb}(f)$ is the set $\{f(Ax + b) : A \in \text{GL}(n, \mathbb{F}) \text{ and } b \in \mathbb{F}^n\}$. 

\[
\begin{bmatrix}
x_1 \\
\vdots \\
x_n
\end{bmatrix}
\]
Orbits

• **Orbit of a polynomial**: For $f \in \mathbb{F}[x_1, ..., x_n]$, the orbit of $f$, denoted by $\text{orb}(f)$ is the set \{f(Ax + b) : A \in \text{GL}(n, \mathbb{F}) and b \in \mathbb{F}^n\}. 

Zero set of $f(x)$

Zero set of $g(x)$
Orbits

• **Orbit of a polynomial**: For $f \in \mathbb{F}[x_1, ..., x_n]$, the orbit of $f$, denoted by $\text{orb}(f)$ is the set $\{f(Ax + b): A \in \text{GL}(n, \mathbb{F}) \text{ and } b \in \mathbb{F}^n\}$.

• **Orbit of a circuit class**: For a circuit class $\mathcal{C}$, the orbit of $\mathcal{C}$, denoted by $\text{orb}(\mathcal{C})$ is the union of $\text{orb}(f)$ for all $f \in \mathcal{C}$. 

Orbits

• **Orbit of a polynomial:** For $f \in \mathbb{F}[x_1, \ldots, x_n]$, the orbit of $f$, denoted by $\text{orb}(f)$ is the set $\{f(Ax + b) : A \in \text{GL}(n, \mathbb{F}) \text{ and } b \in \mathbb{F}^n\}$.

• **Orbit of a circuit class:** For a circuit class $\mathcal{C}$, the orbit of $\mathcal{C}$, denoted by $\text{orb}(\mathcal{C})$ is the union of $\text{orb}(f)$ for all $f \in \mathcal{C}$.

• Recently [MS21, ST21, BG21] studied black-box PIT for orbits of various circuit classes.

MS21: Medini-Shpilka, CCC, 21.
The Power of Orbit Closures

• \( \text{orb}(C) \) is the set of all polynomials that are “well approximated” by polynomials in \( \text{orb}(C) \).

• Ex. 1. \( \text{orb}(\Sigma\Pi) \) contains depth 3 circuits.
The Power of Orbit Closures

• $\text{orb}(C)$ is the set of all polynomials that are “well approximated” by polynomials in $\text{orb}(C)$.

• Ex. 2. $\text{orb}(\text{ROF})$ contains arithmetic formulas.
The Power of Orbit Closures

- $\text{orb}(\mathcal{C})$ is the set of all polynomials that are “well approximated” by polynomials in $\text{orb}(\mathcal{C})$.

- Ex. 3. Iterated Matrix Multiplication $\text{IMM}_{w,d}$. 
The Power of Orbit Closures

• \( \text{orb}(\mathcal{C}) \) is the set of all polynomials that are “well approximated” by polynomials in \( \text{orb}(\mathcal{C}) \).

• Ex. 3. Iterated Matrix Multiplication \( \text{IMM}_{w,d} \).

the \((1, 1)\)-th entry of

\[
\begin{bmatrix}
  x_{1,1,1} & \cdots & x_{1,1,w} \\
  \vdots & \ddots & \vdots \\
  x_{1,w,1} & \cdots & x_{1,w,w}
\end{bmatrix}
\cdots
\begin{bmatrix}
  x_{d,1,1} & \cdots & x_{d,1,w} \\
  \vdots & \ddots & \vdots \\
  x_{d,w,1} & \cdots & x_{d,w,w}
\end{bmatrix}.
\]
The Power of Orbit Closures

• \( \text{orb}(C) \) is the set of all polynomials that are “well approximated” by polynomials in \( \text{orb}(C) \).

• Ex. 3. Iterated Matrix Multiplication \( \text{IMM}_{w,d} \).

• Every polynomial computed by a size \( s \) [formula] is in \( \text{orb} \left( \text{IMM}_{3,poly(s)} \right) \).

• Every polynomial computed by a size \( s \) [Algebraic Branching Program (ABP)] is in \( \text{orb} \left( \text{IMM}_{s,s} \right) \).
The Power of Orbit Closures

• PIT for orbit closures of simple models \( \implies \) PIT for general models like formulas, ABPs, and circuits.

• As a first step, it is natural to try to do PIT for orbits.
PIT for Orbits

• [KS19] gave polynomial time black box PIT for $\text{orb}(\sum_{i \in [n]} x_i^d)$.

• [MS21] gave polynomial time black box PIT for orbit of the continuant polynomial. Orbit closure of the continuant contains all polynomial sized formulas.

MS21: Medini-Shpilka, CCC, 21.
PIT for Orbits

• [KS19] gave polynomial time black box PIT for $\text{orb}(\sum_{i \in [n]} x_i^d)$.

• [MS21] gave polynomial time black box PIT for orbit of the continuant polynomial. Orbit closure of the continuant contains all polynomial sized formulas.

Trace of

$$\begin{bmatrix} x_1 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} x_n & 1 \\ 1 & 0 \end{bmatrix}.$$
PIT for Orbits

• [KS19] gave polynomial time black box PIT for $\text{orb}(\sum_{i \in [n]} x_i^d)$.
• [MS21] gave polynomial time black box PIT for orbit of the continuant polynomial. Orbit closure of the continuant contains all polynomial sized formulas.
• [MS21] gave quasi-polynomial time black box PIT for $\text{orb}(\Sigma \Pi)$.
• [MS21, ST21] gave quasi-polynomial time black box PIT for $\text{orb}(\text{ROF})$.
• [ST21, BG21] gave quasi-polynomial time black box PIT for orbits of commutative ROABPs and constant width ROABPs computing polynomials with individual degree $O(\log n)$.

MS21: Medini-Shpilka, CCC, 21.
Some open problems

• Polynomial time PIT for $\Sigma^k \Pi \Sigma \Pi^\delta$ circuits by proving the Sylvester-Gallai type conjecture proposed by [Gup14].

• Polynomial time black box PIT for ROABPs.

• Black box PIT for $\text{orb}(\text{IMM}_{w,d})$ and orbits of ROABPs.

Gup14: Gupta, ECCC, 14.
Thank You!