Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 89 (2016)

Randomized Polynomial Time Identity Testing for Noncommutative Circuits

V. Arvind[*] Partha Mukhopadhyay[†] S. Raja[‡]

June 6, 2016

Abstract

In this paper we show that the black-box polynomial identity testing for noncommutative polynomials $f \in \mathbb{F}\langle z_1, z_2, \cdots, z_n\rangle$ of degree D and sparsity t, can be done in randomized $\mathrm{poly}(n, \log t, \log D)$ time. As a consequence, if the black-box contains a circuit C of size s computing $f \in \mathbb{F}\langle z_1, z_2, \cdots, z_n\rangle$ which has at most t non-zero monomials, then the identity testing can be done by a randomized algorithm with running time polynomial in s and n and $\log t$. This makes significant progress on a question that has been open for over ten years.

The earlier result by Bogdanov and Wee [BW05], using the classical Amitsur-Levitski theorem, gives a randomized polynomial-time algorithm only for circuits of polynomially bounded syntactic degree. In our result, we place no restriction on the degree of the circuit.

Our algorithm is based on automata-theoretic ideas introduced in [AMS08, AM08]. In those papers, the main idea was to construct deterministic finite automata that isolate a single monomial from the set of nonzero monomials of a polynomial f in $\mathbb{F}\langle z_1, z_2, \cdots, z_n\rangle$. In the present paper, since we need to deal with exponential degree monomials, we carry out a different kind of monomial isolation using nondeterministic automata.

1 Introduction

Noncommutative computation, introduced in complexity theory by Hyafil [Hya77] and Nisan [Nis91], is a central field of algebraic complexity theory. The main algebraic structure of interest is the free noncommutative ring $\mathbb{F}\langle Z\rangle$ over a field $\mathbb{F}$, where $Z = \{z_1, z_2, \cdots, z_n\}$, $z_i, 1 \le i \le n$ are free noncommuting variables.

One of the main problems in the subject is noncommutative Polynomial Identity Testing. The problem can be stated as follows:

Let $f \in \mathbb{F}\langle Z\rangle$ be a polynomial represented by a noncommutative arithmetic circuit C. The polynomial f can be either given by a black-box for C (using which we can evaluate C on matrices with entries from $\mathbb{F}$ or an extension field), or the circuit C may be explicitly given. The algorithmic problem is to check if the polynomial computed by C is identically zero.

We recall the formal definition of a noncommutative arithmetic circuit.

[*] Institute of Mathematical Sciences, Chennai, India, email: arvind@imsc.res.in

[†] Chennai Mathematical Institute, Chennai, India, email: partham@cmi.ac.in

[‡] Institute of Mathematical Sciences, Chennai, India, email: rajas@imsc.res.in

ISSN 1433-8092

Definition 1. *A* noncommutative arithmetic circuit *C over a field $\mathbb{F}$ and indeterminates $z_1, z_2, \cdots, z_n$ is a directed acyclic graph (DAG) with each node of indegree zero labeled by a variable or a scalar constant from $\mathbb{F}$: the indegree 0 nodes are the input nodes of the circuit. Each internal node of the DAG is of indegree two and is labeled by either a $+$ or a $\times$ (indicating that it is a plus gate or multiply gate, respectively). Furthermore, the two inputs to each $\times$ gate are designated as left and right inputs which is the order in which the gate multiplication is done. A gate of C is designated as* output*. Each internal gate computes a polynomial (by adding or multiplying its input polynomials), where the polynomial computed at an input node is just its label. The* polynomial computed *by the circuit is the polynomial computed at its output gate. An arithmetic circuit is a formula if the fan-out of every gate is at most one.*

Notice that if the size of circuit C is s the degree of the polynomial computed by C can be 2^s. In the earlier result [1] by Bogdanov and Wee [BW05], a randomized polynomial-time algorithm was shown for the case when the degree of the circuit C is polynomially bounded in s and n [BW05]. The idea of the algorithm is based on a classical result of Amitsur-Levitski [AL50]. We recall below that part of the Amitsur-Levitski theorem that is directly relevant to polynomial identity testing.

Theorem 1 (Amitsur-Levitski Theorem)**.** *For any field $\mathbb{F}$ (of size more than $2d-1$), a nonzero noncommutative polynomial $P \in \mathbb{F}\langle Z\rangle$ of degree $2d-1$ cannot be a polynomial identity for the matrix algebra $\mathbb{M}_d(\mathbb{F})$. I.e. f does not vanish on all $d \times d$ matrices over $\mathbb{F}$.*

Bogdanov and Wee's randomized PIT algorithm [BW05] applies the above theorem to obtain a randomized PIT as follows: Let $C(z_1, z_2, \cdots, z_n)$ be a circuit of syntactic degree bounded by $2d-1$. For each $i \in [n]$, substitute the variable z_i by a $d \times d$ matrix M_i of commuting indeterminates. More precisely, the $(\ell, k)^{th}$ entry of M_i is $z_{\ell,k}^{(i)}$ where $1 \leq \ell, k \leq d$. By Theorem 1, the matrix $M_f = f(M_1, M_2, \ldots, M_n)$ is not identically zero. Hence, in M_f there is an entry (ℓ', k') which has the *commutative* nonzero polynomial $g_{\ell',k'}$ over the variables $\{z_{\ell,k}^{(i)} : 1 \leq i \leq n, 1 \leq \ell, k \leq d\}$. Notice that the degree of the polynomial $g_{\ell',k'}$ is at most $2d-1$. If we choose an extension field of $\mathbb{F}$ of size at least $4d$, then we get a randomized polynomial identity testing algorithm by the standard Schwartz-Zippel-Lipton-DeMello Lemma [Sch80, Zip79, DL78].

The problem with this approach for general noncommutative circuits (whose degree can be 2^s) is that the dimension of the matrices grows linearly with the degree of the polynomial. Therefore, this approach only yields a randomized exponential time algorithm for the problem. It cannot yield an efficient algorithm for polynomial identity testing for a general noncommutative circuit where the syntactic degree could be exponential in the size of the circuit. Finding an efficient randomized identity test for general noncommutative circuits was a well-known open problem. In this work we resolve it for the case when the polynomial is promised to have the number of non-zero monomials at most exponential in the input size.

2 Main Result

The crux of our result is the following theorem that we show about noncommutative identities which is of independent mathematical interest.

[1] We also note here that Raz and Shpilka [RS05] gives a white-box deterministic polynomial-time identity test for noncommutative algebraic branching programs (ABPs). The result of Forbes-Shpilka [FS13] and Agrawal et al., [AGKS15] gives a quasi-polynomial time black-box algorithm for small degree ABPs.

Theorem 2. *Let $\mathbb{F}$ be a field of size more than d. Let $f \in \mathbb{F}\langle z_1, z_2, \ldots, z_n\rangle$ be a nonzero polynomial of degree d and with t nonzero monomials. Then f cannot be a polynomial identity for the matrix ring $\mathbb{M}_k(\mathbb{F})$ for $k = \log t + 1$.*

The randomized polynomial identity test for noncommutative arithmetic circuits is an immediate corollary. To see this, suppose C is a noncommutative arithmetic circuit of size s computing a polynomial with at most t monomials. The degree of the polynomial f computed by the circuit is bounded by 2^s. Thus, if f is not identically zero, by Theorem 2, the polynomial f does not vanish if we substitute for each z_i, $(\log t + 1) \times (\log t + 1)$ matrices of indeterminates (all distinct). Indeed, f will evaluate to an $(\log t + 1) \times (\log t + 1)$ matrix whose entries are polynomials in commuting variables of degree at most 2^s [2]. For each entry of this matrix, we can employ the standard Schwartz-Zippel-Lipton-DeMello [Sch80, Zip79, DL78] lemma based algorithm for commutative polynomials (by evaluating them over $\mathbb{F}$ or a suitable extension field). This proves the main result of the paper.

Corollary 1. *Let C be a noncommutative circuit of size s given as a black-box computing a polynomial $f \in \mathbb{F}\langle z_1, z_2, \ldots, z_n\rangle$ with sparsity t. Then there is a randomized algorithm to check whether f is an identically zero polynomial and the algorithm runs in time* $\mathrm{poly}(s, n, \log t)$.

Remark 1. *It is interesting to compare Theorem 2 with the classical Amitsur-Levitski theorem. Our result brings out the importance of the number of monomials in a polynomial identity for $d \times d$ matrices. It implies that any polynomial identity f for $d \times d$ matrices over a field $\mathbb{F}$ of size more than* $\deg f$ *must have more than 2^{d-1} monomials.*

We first describe the basic steps required for the proof of Theorem 2. Since we are working in the free noncommutative ring $\mathbb{F}\langle z_1, z_2, \ldots, z_n\rangle$, notice that monomials are free words over the alphabet $\{z_1, z_2, \ldots, z_n\}$, and the polynomial f is an $\mathbb{F}$-linear combination of monomials.

Converting to a bivariate polynomial

It is convenient to convert the given noncommutative polynomial into a noncommutative polynomial in $\mathbb{F}\langle x_0, x_1\rangle$, where x_0 and x_1 are two noncommuting variables. Let

$$f = \sum_{i=1}^{t} c_i w_i$$

with $c_i \in \mathbb{F}$, where w_i are the nonzero monomials (over $\{z_1, z_2, \ldots, z_n\}$) of f. We use the bivariate substitution $\forall i \in [n] : z_i \to x_0 x_1^i x_0$ to encode the words over two variables x_0, x_1. By abuse of notation, we write the resulting polynomial as $f(x_0, x_1) \in \mathbb{F}\langle x_0, x_1\rangle$. Since the above encoding of monomials is bijective, the following claim clearly holds.

Claim 1. *The bivariate noncommutative polynomial $f(x_0, x_1)$ is nonzero if and only if the original polynomial $f \in \mathbb{F}\langle z_1, z_2, \ldots, z_n\rangle$ is nonzero.*

The degree D of $f(x_0, x_1)$ is clearly bounded by $(n + 2)d$.

[2] Note that in general a noncommutative circuit of size s can compute a polynomial that can have 2^{2^s} monomials. For example the polynomial $f(x, y) = (x + y)^{2^s}$ has noncommutative circuit of size $O(s)$ but the number of monomials is 2^{2^s}. Our algorithm can not handle such cases.

Definition 2. *Let $\mathcal{M} \subseteq \{x_0, x_1\}^D$ be a finite set of degree D monomials over variables $\{x_0, x_1\}$. A subset of indices $I \subseteq [D]$ is said to be an* isolating index set *for $\mathcal{M}$ if there is a monomial $m \in \mathcal{M}$ such that for each $m' \neq m, m' \in \mathcal{M}$, there is some index $i \in I$ for which $m[i] \neq m'[i]$. I.e. no other monomial in $\mathcal{M}$ agrees with monomial m on all positions in the index set I.*

The following lemma says that every subset of monomials $\mathcal{M} \subseteq \{x_0, x_1\}^D$ has an isolating index set of size $\log|\mathcal{M}|$. The proof is a simple halving argument.

Lemma 1. *Let $\mathcal{M} \subseteq \{x_0, x_1\}^D$ be a finite set of degree D monomials over variables $\{x_0, x_1\}$. Then $\mathcal{M}$ has an isolating index set of size k which is bounded by $\log|\mathcal{M}|$.*

Proof. The monomials $m \in \mathcal{M}$ are seen as indexed from left to right, where $m[i]$ denotes the variable in the i^{th} position of m. Let $i_1 \leq D$ be the first index such that not all monomials agree on the i^{th} position. Let

$$\begin{aligned} S_0 &= \{m : m[i_1] = x_0\} \\ S_1 &= \{m : m[i_1] = x_1\}. \end{aligned}$$

Either $|S_0|$ or $|S_1|$ is of size at most $|\mathcal{M}|/2$. Let S_{b_1} denote that subset, $b_1 \in \{0, 1\}$. We replace the monomial set $\mathcal{M}$ by S_{b_1} and repeat the same argument for at most $\log|\mathcal{M}|$ steps. Clearly, by this process we identify a set of indices $I = \{i_1, \ldots, i_k\}$, $k \leq \log|\mathcal{M}|$ such that the set shrinks to a singleton set $\{m\}$. Clearly, I is an isolating index set as witnessed by the *isolated monomial m.* □

Remark 2. *Notice that the size of the isolating index set denoted k is bounded by $\log t$ as well as the degree D of the polynomial $f(x_0, x_1)$.*

NFA construction

In our earlier paper [AMS08] (for sparse polynomial identity testing) we used a deterministic finite state automaton to isolate a monomial by designing an automaton which accepts a unique monomial. This will not work for the proof of Theorem 2 because the number of states that such a deterministic automaton requires is the length of the monomial which could be exponentially large. It turns out that we can use a small *nondeterministic* finite automaton which will guess the isolating index set for the set of nonzero monomials of f. The complication is that there are exponentially many wrong guesses. However, it turns out that if we make our NFA a *substitution automaton*, we can ensure that the monomials computed on different nondeterministic paths (which correspond to different guesses of the isolating index set) all have disjoint support. Once we have this property, it is easy to argue that for the correct nondeterministic path, the computed commutative polynomial is nonvanishing (because the isolated monomial cannot be cancelled). With this intuition, we proceed with the simple technical details.

We describe the construction of a substitution NFA that substitutes, on its transition edges, a new commuting variable for the variable (x_0 or x_1) that it reads. Formally, let A denote the NFA given by a 5-tuple $A = \langle Q, \Sigma = \{x_0, x_1\}, \delta, q_o, q_f \rangle$, where $Q = \{q_0, q_1, q_2, \ldots, q_k\}$ and $q_f = q_k$. We use the indices $i_1, \ldots, i_k$ from Lemma 1 to define the transition of A. The set of indices partition each monomial m into $k + 1$ blocks as follows.

$$m[1, i_1 - 1]m[i_1]m[i_1 + 1, i_2 - 1]m[i_2] \cdot \cdots \cdot m[i_{k-1} + 1, i_{k-1}]m[i_k]m[i_{k+1}, D],$$

where $m[i]$ denotes the variable in i^{th} position of m and $m[i,j]$ denotes the submonomial of m from positions i to j.

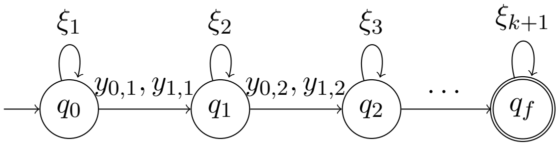
We use a new set of variables for different blocks and the indices $i_1, \ldots, i_k$ as follows. The *block variables* are $\bigcup_{j\in[k+1]}\{\xi_j\}$, and the *index variables* are $\bigcup_{j\in[k]}\{y_{0,j}, y_{1,j}\}$.

Now we are ready to describe the transitions of the automaton. When the NFA is reading the input variables in block j, it will replace each $x_b, b \in \{0,1\}$ by block variable ξ_j. Then the NFA nondeterministically decides if block j is over and the current location is an index in the isolating set. In that case, the NFA replaces the variable x_b that is read by the index variable $y_{b,j}$ and the NFA also increments the block number to $j+1$. It will now make its transitions in the $(j+1)^{st}$ block as described above.

The NFA is formally described by the following simple transition rules. For $0 \leq i \leq k-1$, and $b \in \{0,1\}$,

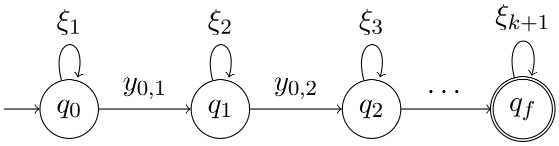
$$\delta(q_i, x_b) \xrightarrow{\xi_{i+1}} q_i$$
$$\delta(q_i, x_b) \xrightarrow{y_{b,i+1}} q_{i+1}.$$

We depict the description of the automaton in the following figure.



Clearly, the transitions of the automaton A can be described by two $(k+1)\times(k+1)$ adjacency matrices M_{x_0} and M_{x_1} corresponding to the moves of the automaton on input x_0 and input x_1.

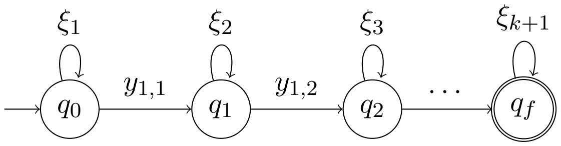
More precisely, for variable x_0, we take the adjacency matrix M_{x_0} of the following labeled directed graph extracted from the above automaton.



The corresponding matrix M_{x_0} of dimension $(k+1)\times(k+1)$, we substitute for x_0 is the following.

$$\mathbf{M_{x_0}} = \begin{pmatrix} \xi_1 & y_{0,1} & 0 & \ldots & 0 & 0 \\ 0 & \xi_2 & y_{0,2} & \ldots & 0 & 0 \\ 0 & 0 & \xi_3 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & \xi_k & y_{0,k} \\ 0 & 0 & 0 & \ldots & 0 & \xi_{k+1} \end{pmatrix}$$

Similarly, for variable x_1 we take the adjacency matrix M_{x_1} of the following labeled directed graph.



The corresponding matrix M_{x_1} of dimension $(k+1)\times(k+1)$, we substitute for x_1 is the following.

$$\mathbf{M_{x_1}} = \begin{pmatrix} \xi_1 & y_{1,1} & 0 & \dots & 0 & 0 \\ 0 & \xi_2 & y_{1,2} & \dots & 0 & 0 \\ 0 & 0 & \xi_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \xi_k & y_{1,k} \\ 0 & 0 & 0 & \dots & 0 & \xi_{k+1} \end{pmatrix}$$

The rows and the columns of the matrices M_{x_0} and M_{x_1} are indexed by the states of the automaton and the entries are either block variables or index variables as indicated in the transition diagram. Let

$$f = \sum_{i=1}^{t} c_i w_i.$$

Define the matrix $w_i(M_{x_0}, M_{x_1})$ obtained by substituting in w_i the matrix M_{x_b} for $x_b, b \in \{0,1\}$ and multiplying these matrices. The following proposition is immediate as f is a linear combination of the w_i's.

Proposition 1. $M_f = f(M_{x_0}, M_{x_1}) = \sum_{i=1}^{t} c_i w_i(M_{x_0}, M_{x_1})$.

Now we are ready to prove Theorem 2.

Proof of Theorem 2. We assume that n-variate polynomial f is converted to the bivariate polynomial $f(x_0, x_1)$ over x_0 and x_1. Let $\mathcal{M}$ denote the set of nonzero monomials of degree D occurring in f, where D is the degree of f. Then we can write the polynomial $f = \sum_{j=1}^{t} c_j w_j$ in two parts

$$f = \sum_{w_j \in \mathcal{M}} c_j w_j + \sum_{w_j \notin \mathcal{M}} c_j w_j,$$

where $\sum_{w_j \in \mathcal{M}} c_j w_j$ is the homogeneous degree D part of f.

Let us assume, without loss of generality, that w_1 is in $\mathcal{M}$ and it is the monomial isolated in Lemma 1, and the isolating index set be $I = \{i_1, i_2, \cdots, i_k\}$ such that for all $w_j \in \mathcal{M}$, $w_j|_I \neq w_1|_I$ (i.e. the projections of each $w_j, j \neq 1$ on index set I differs from the projection of w_1). Let

$$w_1 = x_{b_1} x_{b_2} \cdots x_{b_D},$$

where $b_j \in \{0,1\}$.

The following claim is immediate.

Claim 2. *For each index set $J = \{j_1, j_2, \cdots, j_k\}$ nondeterministically picked by the substitution NFA, each nonzero degree D monomial w_j occurring in f is transformed into a unique degree*

D monomial $w_{j,J}$ (which is over the block and index variables). More precisely, let $\xi_J = \xi_1^{j_1-1}\xi_2^{j_2-j_1}\cdots\xi_{k+1}^{D-j_k}$ and $y_{j,J} = y_{a_1,1}y_{a_2,2}\cdots y_{a_k,k}$. Then

$$w_{j,J} = \xi_J y_{j,J}.$$

Notice that for two distinct index sets J and J' we clearly have $\xi_J \neq \xi_{J'}$. We also note that $y_{j,J}$ is essentially the projection of the degree D monomial w_j to the index set J; if variables x_b occurs in the j_k^{th} position of w_j then it is replaced by $y_{b,k}$ in $y_{j,J}$.

Furthermore, for each $w_j \in \mathcal{M}$ we note that the $(q_o, q_f)^{th}$ entry of the matrix $w_j(M_{x_0}, M_{x_1})$ is the sum $\sum_J w_{j,J} = \sum_J \xi_J y_{j,J}$. For different index sets J the monomials $\xi_J y_{j,J}$ are all distinct.

Let f_J be the polynomial

$$f_J = \sum_{j=1}^{t} c_j w_{j,J} = \sum_{w_j \in \mathcal{M}} c_j w_{j,J} + \sum_{w_j \notin \mathcal{M}} c_j w_{j,J}.$$

Claim 3. *After the matrix substitution $x_0 = M_{x_0}$ and $x_1 = M_{x_1}$ in the polynomial f we note that the $(q_0, q_f)^{th}$ entry of the matrix $f(M_{x_0}, M_{x_1})$ is $\sum_J f_J$.*

The above claim clearly holds because the polynomial f_J is the contribution of the nondeterministic path corresponding to index set J.

Claim 4. *For any two index sets J, J' and any monomial $w_j \in \mathcal{M}$, the corresponding commutative monomials $w_{j,J}$ and $w_{j,J'}$ are distinct.*

To see this claim it suffices to see that $w_{j,J} = \xi_J y_{j,J}$ and $w_{j,J'} = \xi_{J'} y_{j,J'}$ and we have already observed that $\xi_J \neq \xi_{J'}$.

Finally, we focus on the monomial $w_{1,I}$ occurring in the polynomial $\sum_J f_J$, where w_1 is the isolated monomial and I is the isolated index set.

Claim 5. *The coefficient of $w_{1,I}$ in the polynomial $\sum_J f_J$ is c_1. As a consequence, the polynomial $\sum_J f_J$ which occurs in the (q_0, q_f) entry of the matrix $M_f = f(M_{x_0}, M_{x_1})$ is nonzero because the coefficient of $w_{1,I}$ in it is nonzero.*

To see the above claim we note the following points:

1. For the monomials $w_j \notin \mathcal{M}$ notice that for each index set J of size k, the contribution to the $(q_0, q_f)^{th}$ entry of the matrix $f(M_{x_0}, M_{x_1})$ is a monomial of degree $\deg(w_j)$, and $\deg(w_j) < D$. Hence, these monomials have no influence on the coefficient of $w_{1,I}$ in the polynomial $\sum_J f_J$.

2. Consider monomials $w_j \in \mathcal{M}$ for $j \neq 1$. Notice that

$$w_{1,I} = \xi_I y_{1,I},$$

and for $j \neq 1$

$$w_{j,I} = \xi_I y_{j,I},$$

and the monomials $y_{1,I}$ and $y_{j,I}$ are different because I is an isolating index set and the monomial w_1 is isolated. I.e. the monomials $w_{1,I}$ and $w_{j,I}$ will necessarily differ in the index variables occurring in them as a consequence of the isolation property.

Therefore, the monomials $w_j \in \mathcal{M}$ for $j \neq 1$ also have no influence on the coefficient of $w_{1,I}$ in the polynomial $\sum_J f_J$.

Hence, we conclude that the $(q_0, q_f)^{th}$ entry of the matrix $M_f = f(M_{x_0}, M_{x_1})$ is a nonzero polynomial $\sum_J f_J$ in the commuting variables $\bigcup_{j\in[k+1]}\{\xi_j\}$ and $\bigcup_{j\in[k]}\{y_{0,j}, y_{1,j}\}$. Moreover the degree of polynomial $\sum_J f_J$ is D. Now we can apply Schwartz-Zippel-Lipton-DeMello Lemma [Sch80, Zip79, DL78] to conclude that the polynomial $\sum_J f_J$ will be nonzero over a suitable extension of size more than $(n+2)d$ of the field $\mathbb{F}$. Since the polynomial f is nonzero over the algebra $\mathbb{M}_{k+1}(\mathbb{F})$, it is also nonzero over the algebra $\mathbb{M}_{\log t+1}(\mathbb{F})$. This completes the proof of Theorem 2. □

Acknowledgement

The number of monomials in a noncommutative polynomial which has an arithmetic circuit of size s can actually be doubly exponential in s. We thank Srikanth Srinivasan for pointing this out to us.

References

[AGKS15] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena, *Hitting-sets for ROABP and sum of set-multilinear circuits*, SIAM J. Comput. **44** (2015), no. 3, 669–697.

[AL50] Avraham Shimshon Amitsur and Jacob Levitzki, *Minimal identities for algebras*, Proceedings of the American Mathematical Society **1** (1950), no. 4, 449–463.

[AM08] Vikraman Arvind and Partha Mukhopadhyay, *Derandomizing the isolation lemma and lower bounds for circuit size*, Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques, 11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008, Boston, MA, USA, August 25-27, 2008. Proceedings, 2008, pp. 276–289.

[AMS08] Vikraman Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan, *New results on noncommutative and commutative polynomial identity testing*, Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA, 2008, pp. 268–279.

[BW05] Andrej Bogdanov and Hoeteck Wee, *More on noncommutative polynomial identity testing*, 20th Annual IEEE Conference on Computational Complexity (CCC 2005), 11-15 June 2005, San Jose, CA, USA, 2005, pp. 92–99.

[DL78] Richard A. DeMillo and Richard J. Lipton, *A probabilistic remark on algebraic program testing*, Inf. Process. Lett. **7** (1978), no. 4, 193–195.

[FS13] Michael A. Forbes and Amir Shpilka, *Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs*, 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA, 2013, pp. 243–252.

[Hya77] Laurent Hyafil, *The power of commutativity*, 18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977, 1977, pp. 171–174.

[Nis91] Noam Nisan, *Lower bounds for non-commutative computation (extended abstract)*, STOC, 1991, pp. 410–418.

[RS05] Ran Raz and Amir Shpilka, *Deterministic polynomial identity testing in non-commutative models*, Computational Complexity **14** (2005), no. 1, 1–19.

[Sch80] J. T. Schwartz, *Fast probabilistic algorithms for verification of polynomial identities*, J. ACM **27** (1980), no. 4, 701–717.

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