Non-Commutative Formulas and Frege Lower Bounds: a New Characterization of Propositional Proofs

Fu Li[1], Iddo Tzameret[2], and Zhengyu Wang[3]

1 **Institute for Interdisciplinary Information Sciences, Tsinghua University***
2 **Department of Computer Science, Royal Holloway, University of London**
`Iddo.Tzameret@rhul.ac.uk`
3 **Department of Computer Science, Harvard University**

Abstract

Does every Boolean tautology have a short propositional-calculus proof? Here, a propositional-calculus (i.e. Frege) proof is any proof starting from a set of axioms and deriving new Boolean formulas using a fixed set of sound derivation rules. Establishing any super-polynomial size lower bound on Frege proofs (in terms of the size of the formula proved) is a major open problem in proof complexity, and among a handful of fundamental hardness questions in complexity theory by and large. Non-commutative arithmetic formulas, on the other hand, constitute a quite weak computational model, for which exponential-size lower bounds were shown already back in 1991 by Nisan [20], using a particularly transparent argument.

In this work we show that Frege lower bounds in fact follow from corresponding size lower bounds on non-commutative formulas computing certain polynomials (and that such lower bounds on non-commutative formulas must exist, unless ***NP=coNP***). More precisely, we demonstrate a natural association between tautologies T to non-commutative polynomials p, such that:

- if T has a polynomial-size Frege proof then p has a polynomial-size non-commutative arithmetic formula; and conversely, when T is a DNF, if p has a polynomial-size non-commutative arithmetic formula over $GF(2)$ then T has a Frege proof of quasi-polynomial size.

The argument is a characterization of Frege proofs as non-commutative formulas: we show that the Frege system is (quasi-) polynomially equivalent to a *non-commutative Ideal Proof System* (IPS), following the recent work of Grochow and Pitassi [10] that introduced a propositional proof system in which proofs are arithmetic circuits, and the work in [35] that considered adding the commutator as an axiom in algebraic propositional proof systems. This gives a characterization of propositional Frege proofs in terms of (non-commutative) arithmetic formulas that is tighter than (the formula version of IPS) in Grochow and Pitassi [10], in the following sense:

(i) The non-commutative IPS is *polynomial-time checkable* – whereas the original IPS was checkable in probabilistic polynomial-time; and

(ii) Frege proofs *unconditionally* quasi-polynomially simulate the non-commutative IPS – whereas Frege was shown to efficiently simulate IPS only assuming that the decidability of PIT for (commutative) arithmetic formulas by polynomial-size circuits is efficiently provable in Frege.

1998 ACM Subject Classification F.2.2 [Analysis of Algorithms and Problem Complexity] Non-numerical Algorithms and Problems – Complexity of proof procedures

Keywords and phrases Proof complexity, algebraic complexity, non-commutative formulas, Frege proofs, ideal proof system

Digital Object Identifier 10.4230/LIPIcs.CCC.2015.412

---

* The first author was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003, 61373002.

30th Conference on Computational Complexity (CCC'15).
Editor: David Zuckerman; pp. 412–432
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Propositional proof complexity

The field of propositional proof complexity aims to understand and analyze the computational resources required to prove propositional statements. The problems the field poses are fundamental, difficult and go back to the work of Cook and Reckhow [8], who showed the immediate relevance of these problems to the $\boldsymbol{NP}$ vs. $\boldsymbol{coNP}$ problem (and thus the $\boldsymbol{P}$ vs. $\boldsymbol{NP}$ problem).

Among the major unsolved questions in proof complexity, is whether the standard propositional logic calculus, either in the form of the Sequent Calculus, or equivalently, in the axiomatic form of Hilbert proofs (i.e., Frege proofs), is polynomially bounded; that is, whether every propositional tautology (or unsatisfiable formula) has a proof whose size is polynomially bounded (refutation, resp.) in the size of the formula proved. Here, we consider the size of proofs as the number of symbols it takes to write them down, where each formula in the proof is written as a Boolean *formula* (in other words we count the total number of logical gates appearing in the proof where each proof-line is a formula). It is known [29] that all Frege proof-systems (formally, a Frege proof system is any propositional proof system with a fixed number of axiom schemes and sound derivation rules that is also implicationally complete, and in which proof-lines are written as propositional formulas (see e.g., [14] and Definition 2.4 below)) as well as the Gentzen sequent calculus (with the cut rule) are polynomially equivalent to each other, and hence it does not matter precisely which rules, axioms, and logical-connectives we use.

Complexity-wise, the Frege proof system is considered a very strong system alas a poorly understood one. The qualification *strong* here has several meanings: first, that no super-polynomial lower bound is known for Frege proofs. Second, that there are not even good hard candidates for the Frege system (see [4, 17, 18] for a further discussion on hard proof complexity candidates). Third, that for most hard instances (e.g., the pigeonhole principle and Tseitin tautologies) that are known to be had for weaker systems (e.g., resolution, cutting planes, etc.), there *are* known polynomial bounds on Frege proofs. Fourth, that proving super-polynomial lower bounds on Frege proofs seems to a certain extent out of reach of current techniques. And finally, that by the common (mainly informal) correspondence between circuits and proofs – namely, the correspondence between a circuit-class $\mathcal{C}$ and a proof system in which every proof-line is written as a circuit from $\mathcal{C}$ (to be more precise, one has to associate a circuit class $\mathcal{C}$ with a proof system in which a *family* of proofs is written such that every proof-line in the family is a circuit family from $\mathcal{C}$) – Frege system corresponds to the circuit class of polynomial-size $\log(n)$-depth circuits denoted $\boldsymbol{NC}^1$ (equivalently, of polynomial-size formulas [32]), considered to be a strong computational model for which no (explicit) super-polynomial lower bounds are currently known.

Accordingly, proving lower bounds on Frege proofs is considered an extremely hard task. In fact, the best lower bound known today is only quadratic [14], which uses a fairly simple syntactic argument. If we put further impeding restrictions on Frege proofs, like restricting the depth of each formula appearing in a proof to a certain fixed constant, exponential lower bounds can be obtained [1, 21, 21]. Although these constant-depth Frege exponential-size lower bounds go back to Ajtai's result from 1988, they are still in some sense the state-of-the-art in proof complexity lower bounds (beyond the important developments on weaker proof systems, such as resolution and its weak extensions). Constant-depth Frege lower bounds use quite involved probabilistic arguments, mainly specialized switching lemmas tailored for specific tautologies (namely, counting tautologies, most notable of which are the Pigeonhole

Principle tautologies). Even random k-CNF formulas near the satisfiability threshold are not known to be hard for constant-depth Frege (let alone hard for [unrestricted depth] Frege).

All of the above goes to emphasize the importance, basic nature and difficulty in understanding the complexity of strong propositional proof systems, while showing how little is actually known about these systems.

1.2 Prominent directions for understanding propositional proofs

As we already mentioned, there is a guiding line in proof complexity which states a correspondence between the complexity of circuits and the complexity of proofs. This correspondence is mainly informal, but there are seemingly good indications showing it might be more than a superficial analogy. One of the most compelling evidence for this correspondence is that there is a precise formal correspondence (cf. [7]) between the first-order logical theories of bounded arithmetic (whose axioms state the existence of sets taken from a given complexity class $\mathcal{C}$) to propositional proof systems (in which proof-lines are circuits from $\mathcal{C}$).

Another facet of the informal correspondence between circuit complexity and proof complexity is that circuit hardness can sometimes be used to obtain proof complexity hardness. The most notable example of this are the lower bounds on constant-depth Frege proofs mentioned above: constant-depth Frege proofs can be viewed as propositional logic operating with $\boldsymbol{AC}^0$ circuits, and the known lower bounds on constant depth Frege proofs (cf. [1, 16, 21]) use techniques borrowed from $\boldsymbol{AC}^0$ circuits lower bounds. The success in moving from circuit hardness towards proof-complexity hardness has spurred a flow of attempts to obtain lower bounds on proof systems other than constant depth Frege. For example, Pudlák [22] and Atserias et al. [2] studied proofs based on monotone circuits, motivated by known exponential lower bounds on monotone circuits. Raz and Tzameret [28, 27, 34] investigated algebraic proof systems operating with multilinear formulas, motivated by lower bounds on multilinear formulas for the determinant, permanent and other explicit polynomials [24, 23]. Atserias et al. [3], Krajíček [15] and Segerlind [31] have considered proofs operating with ordered binary decision diagrams (OBDDs), and the second author [35] initiated the study of proofs operating with non-commutative formulas (see Sec. 1.5 for a comparison with the current work).

Until quite recently it was unknown whether the correspondence between proofs and circuits is two-sided, namely, whether proof complexity hardness (of concrete known proof systems) can imply any computational hardness. An initial example of such an implication from proof hardness to circuit hardness was given by Raz and Tzameret [28]. They showed that a separation between algebraic proof systems operating with arithmetic circuits and multilinear arithmetic circuits, resp., for an explicit family of polynomials, implies a separation between arithmetic circuits and multilinear arithmetic circuits. In a recent significant development about the complexity of strong proof systems, Grochow and Pitassi [10] demonstrated a much stronger correspondence. They introduced a natural propositional proof system, called the *Ideal Proof System* (IPS for short), for which *any* super-polynomial size lower bound on IPS implies a corresponding size lower bound on arithmetic circuits, and formally, that the permanent does not have polynomial-size arithmetic circuits. The IPS is defined as follows:

► **Definition 1.1** (Ideal Proof System (IPS) [10]). Let $F_1(\overline{x}),\dots,F_m(\overline{x})$ be a system of polynomials in the variables $x_1,\dots,x_n$, where the polynomials $x_i^2-x_i$, for all $1\le i\le n$, are part of this system. An *IPS refutation (or certificate)* that the F_i's polynomials have no 0-1 solutions is a polynomial $C(\overline{x},\overline{y})$ in the variables $x_1,\dots,x_n$ and $y_1,\dots,y_m$, such that:

1. $F(x_1,\dots,x_n,\overline{0})=0$; and
2. $F(x_1,\dots,x_n,F_1(\overline{x}),\dots,F_m(\overline{x}))=1$.

The essence of IPS is that a proof (or refutation) is a *single* polynomial that can be written simply as an arithmetic *circuit* or *formula*. The advantage of this formulation is that now we can obtain direct connections between circuit/formula hardness (i.e., "computational hardness") and hardness of proofs. Grochow and Pitassi showed indeed that a lower bound on IPS written as an arithmetic circuit implies that the permanent does not have polynomial-size algebraic circuits (Valiant's conjectured separation $\mathsf{VNP}\neq\mathsf{VP}$ [36]); And similarly, a lower bound on IPS written as an arithmetic *formula* implies that the permanent does not have polynomial-size algebraic formulas ($\mathsf{VNP}\neq\mathsf{VP}_e$, ibid).

Under certain assumptions, Grochow and Pitassi [10] were able to connect their result to standard propositional-calculus proof systems, i.e., Frege and Extended Frege. Their assumption was the following: *Frege has polynomial-size proofs of the statement expressing that the PIT for arithmetic formulas is decidable by polynomial-size Boolean circuits* (*PIT for arithmetic formulas* is the problem to decide whether an input arithmetic formula computes the (formal) zero polynomial). They showed that under this assumption super-polynomial lower bounds on Frege proofs imply that the permanent does not have polynomial-size arithmetic circuits. This, in turn, can be considered as a (conditional) justification for the apparent difficulty in proving lower bounds on strong proof systems (We focus only on the relevant results about Frege proofs from [10]; and not the results about Extended Frege in [10]; the latter proof system operates, essentially, with Boolean circuits, in the same way that Frege operates with Boolean formulas (equivalently $\boldsymbol{NC}^1$ circuits)).

1.3 Overview of results and proofs

1.3.1 Sketch

In this paper we contribute to the understanding of strong proof systems such as Frege, and to the fundamental search for lower bounds on these systems, by formulating a very natural proof system – a non-commutative variant of the ideal proof system – which we show captures *unconditionally* (up to a quasi-polynomial-size increase) propositional Frege proofs. A proof in the non-commutative IPS is simply a *single non-commutative polynomial* written as a non-commutative formula. This gives a fairly compelling and simple new characterization of the proof complexity of propositional Frege proofs. Moreover, it brings new hope for achieving lower bounds on strong proof systems, by reducing the task of lower bounding Frege proofs to the following seemingly much more manageable task: proving matrix rank lower bounds on the matrices associated with certain non-commutative polynomials (in the sense of Nisan [20]; see below for details).

We also tighten the results in Grochow and Pitassi [10], in the sense that we show that in order to obtain a characterization of Frege proofs in terms of an ideal proof system it is advantageous to consider non-commutative polynomials instead of commutative ones (as well as to add the commutator axioms). This shows that, at least for Frege, and in the framework of the ideal proof system, lower bounds on Frege proofs do not necessarily entail in themselves very strong computational lower bounds.

1.3.2 Some preliminaries: non-commutative polynomials and formulas

A *non-commutative polynomial* over a given field $\mathbb{F}$ and with the variables $X := \{x_1, x_2, \ldots\}$ is a formal sum of monomials with coefficients from $\mathbb{F}$ such that the product of variables is non-commuting. For example, $x_1x_2 - x_2x_1 + x_3x_2x_3^2 - x_2x_3^3$, $x_1x_2 - x_2x_1$ and 0 are three distinct polynomials in $\mathbb{F}\langle X\rangle$. The ring of non-commutative polynomials with variables X with coefficients from $\mathbb{F}$ is denoted $\mathbb{F}\langle X\rangle$.

A *polynomial* (i.e., a *commutative* polynomial) over a field is defined in the same way as a non-commutative polynomial except that now the product of variables *is* commutative; that is, it is a sum of (commutative) monomials.

A *non-commutative arithmetic formula* (non-commutative formula for short; see Definition 2.5) is a fan-in two labeled tree, with edges directed from leaves towards the root, such that the leaves are labeled with field elements (for a given field $\mathbb{F}$) or variables $x_1, \ldots, x_n$, and internal nodes (including the root) are labeled with a plus $+$ or product $\times$ gates. A product gate has an order on its two children (holding the order of non-commutative product). A non-commutative formula computes a non-commutative polynomial in the natural way (see Definition 2.5).

Exponential-size lower bounds on non-commutative formulas (over any field) were established by Nisan [20]. The idea (in retrospect) is quite simple: first transform a non-commutative formula into an algebraic branching program (ABP); and then show that the number of nodes in the ith layer of an ABP computing a degree d homogenous non-commutative polynomial f is bounded from below by the rank of the degree i-partial-derivative matrix of f. (The degree i partial derivative matrix of f is the matrix whose rows are all non-commutative monomials of degree i and columns are all non-commutative monomials of degree $d-i$, such that the entry in row M and column N is the coefficient of the d degree monomial $M \cdot N$ in f.) Thus, lower bounds on non-commutative formulas follow from quite immediate rank arguments (e.g., the partial derivative matrices associated with the permanent and determinant can easily be shown to have high ranks).

1.3.3 Non-commutative ideal proof system

Recall the IPS refutation system in Definition 1.1 above. We use the idea introduced in [35], that considered adding the commutator $x_1x_2 - x_2x_1$ as an axiom in propositional algebraic proof systems, to define a refutation system that polynomially simulates Frege:

▶ **Definition 1.2** (Non-commutative IPS)**.** Let $\mathbb{F}$ be a field. Assume that $F_1(\overline{x}) = F_2(\overline{x}) = \cdots = F_m(\overline{x}) = 0$ is a system of non-commutative polynomial equations from $\mathbb{F}\langle x_1, \ldots, x_n\rangle$, and suppose that the following set of equations (axioms) are included in the $F_i(\overline{x})$'s:

Boolean axioms: $x_i \cdot (1 - x_i)\,,\quad$ for all $1 \le i \le n\,;$

Commutator axioms: $x_i \cdot x_j - x_j \cdot x_i\,,\quad$ for all $1 \le i < j \le n\,.$

Suppose that the $F_i(\overline{x})$'s have no common 0-1 solutions. (One can check that the $F_i(\overline{x})$'s have no common 0-1 solutions in $\mathbb{F}$ iff they do not have a common 0-1 solution in every $\mathbb{F}$-algebra.) A *non-commutative IPS refutation* (or *certificate*) that the system of $F_i(\overline{x})$'s is unsatisfiable is a non-commutative polynomial $\mathfrak{F}(\overline{x}, \overline{y})$ in the variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ (i.e. $\mathfrak{F} \in \mathbb{F}\langle \overline{x}, \overline{y}\rangle$), such that:

1. $\mathfrak{F}(x_1, \ldots, x_n, \overline{0}) = 0$; and
2. $\mathfrak{F}(x_1, \ldots, x_n, F_1(\overline{x}), \ldots, F_m(\overline{x})) = 1$.

We always assume that the non-commutative IPS refutation is written as a *non-commutative formula*. Hence the ***size*** of a non-commutative IPS refutation is the minimal size of a non-commutative formula computing the non-commutative IPS refutation.

The main result of this paper is that the non-commutative IPS (over either $\mathbb{Q}$ or $\mathbb{Z}_q$, for any prime q) polynomially simulates Frege; and conversely, Frege quasi-polynomially simulates the non-commutative IPS (over $GF(2)$). We explain the results in what follows.

For the purpose of the next theorem, we use a standard translation of propositional formulas T into non-commutative arithmetic formulas:

▶ **Definition 1.3.** Let $\mathrm{tr}(x_i) := x_i$, for variables x_i; $\mathrm{tr}(false) := 1$; $\mathrm{tr}(true) := 0$; and by induction on the size of the propositional formula: $\mathrm{tr}(\neg T_1) := 1 - \mathrm{tr}(T_1)$; $\mathrm{tr}(T_1 \vee T_2) = \mathrm{tr}(T_1) \cdot \mathrm{tr}(T_2)$ and finally $\mathrm{tr}(T_1 \wedge T_2) = 1 - ((1 - \mathrm{tr}(T_1)) \cdot (1 - \mathrm{tr}(T_2)))$.

For a non-commutative *formula* f denote by $\widehat{f}$ the non-commutative *polynomial* computed by f. Thus, T is a propositional tautology (i.e., a Boolean formula that is satisfied by every assignment) iff $\widehat{\mathrm{tr}(T)} = 0$ for every 0-1 assignment to the underlying variables of the non-commutative polynomial.

▶ **Theorem 1.4.** *Let $\mathbb{F}$ be either $\mathbb{Q}$ or $\mathbb{Z}_q$, for a prime q. The non-commutative IPS refutation system, when refutations are written as non-commutative formulas over $\mathbb{F}$, polynomially simulates the Frege system. More precisely, for every propositional tautology T, if T has a polynomial-size Frege proof then there is a non-commutative IPS certificate (over $\mathbb{F}$) of* $\mathrm{tr}(\neg T)$ *that has a polynomial non-commutative formula size.*

The proof of Theorem 1.4 proceeds as follows. To simulate Frege proofs we use an intermediate proof system $\mathcal{F}$-$\mathcal{PC}$ formulated by Grigoriev and Hirsch [9]. The $\mathcal{F}$-$\mathcal{PC}$ system (Definition 2.7) is akin to the polynomial calculus refutation system [6], except that we operate in $\mathcal{F}$-$\mathcal{PC}$ with *arithmetic formulas* treated as syntactic terms, instead of writing polynomials throughout the proof as sum of monomials. We have the two rules of polynomial calculus: from a pair of previously derived polynomials f, g we can derive $af + bg$ for $a, b \in \mathbb{F}$, and from f we can derive $x_i \cdot f$, for any variable x_i. We also have local rewriting rules, that can operate on any *sub*-formula of an arithmetic formula appearing in the proof. These rewriting rules express simple operations on polynomials like commutativity of addition and product, associativity, distributivity, etc.

Grigoriev and Hirsch [9] showed that $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates Frege proofs, and that for tree-like Frege proofs the polynomial simulation yields tree-like $\mathcal{F}$-$\mathcal{PC}$ proofs. Since tree-like Frege is polynomially equivalent to Frege (because Frege proofs can always be balanced to depth which is logarithmic in their size; cf. [14] for a proof), we have that tree-like $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates (dag-like) Frege proofs.

To conclude Theorem 1.4 it therefore remains to prove that non-commutative IPS polynomially simulates tree-like $\mathcal{F}$-$\mathcal{PC}$ proofs. This can be proved by induction on the number of lines in the $\mathcal{F}$-$\mathcal{PC}$ proofs. The interesting case in the induction is the simulation of the commutativity rewrite-rule for products by the non-commutative IPS system, which is done using the commutator axioms.

Now, since we write refutations as non-commutative formulas we can use the polynomial-time deterministic Polynomial Identity Testing algorithm for non-commutative formulas, devised by Raz and Shpilka [26], to check in *deterministic* polynomial-time the correctness of non-commutative IPS refutations. Therefore, we obtain:

▶ **Corollary 1.5.** *The non-commutative IPS is a sound and complete Cook-Reckhow refutation system. That is, it is a sound and complete refutation system for unsatisfiable propositional formulas in which refutations can be checked for correctness in deterministic polynomial-time.*

This should be contrasted with the original (commutative) IPS of [10], for which verification of refutations is done in *probabilistic* polynomial time (using the standard Schwartz-Zippel [30, 37] PIT algorithm).

The major consequence of Theorem 1.4 is that to prove a super-polynomial Frege lower bound it is now sufficient to prove a super-polynomial lower bound on non-commutative formulas computing certain polynomials. Specifically, it is enough to prove that any non-commutative IPS certificate $\mathfrak{F}(\overline{x}, \overline{y})$ (which is simply a non-commutative polynomial) has a super-polynomial non-commutative formula size; and yet in another words, it suffices to show that any such $\mathfrak{F}$ must have a super-polynomial total rank according to the associated partial-derivatives matrices discussed before.

We now consider the *other direction*, namely, whether Frege can simulate the non-commutative IPS. We show that it does for CNFs (this is the case considered in [10]), over $GF(2)$, and with only a quasi-polynomial increase in size. For convenience, we use a slightly different translation of clauses to non-commutative formulas than Definition 1.3:

▶ **Definition 1.6.** Given a Boolean formula f we define the non-commutative formula translation $\text{tr}'(f)$ as follows. Let $\text{tr}'(x) := 1 - x$ and $\text{tr}(\neg x) := x$, for x a variable. And let $\text{tr}'(f_1 \vee \ldots \vee f_r) := \text{tr}'(f_1) \cdots \text{tr}'(f_r)$ (where the sequence of products stands for a tree of product gates with $\text{tr}'(f_i)$ as leaves). Further, for a clause k_i in a CNF $\phi = k_1 \wedge k_2 \ldots \wedge k_m$, denote by Q_i^ϕ the non-commutative formula translation $\text{tr}'(k_i)$ of k_i, where $i = 1, 2, \ldots, m$.

▶ **Theorem 1.7.** *For a CNF $\phi = k_1 \wedge \ldots \wedge k_m$ where $Q_1^\phi, \ldots, Q_m^\phi$ are the corresponding non-commutative formulas for the clauses, if there is a non-commutative IPS refutation of size s of $Q_1^\phi, \ldots, Q_m^\phi$ over $GF(2)$, then there is a Frege proof of size $s^{O(\log s)}$ of $\neg\phi$.*

The proof of Theorem 1.7 consists of several separate steps of independent interest. Essentially, the argument is a short Frege proof for a *reflection principle* for the non-commutative IPS system (a reflection principle for a given proof system P is a statement that says that if a formula is provable in P than the formula is also true). The argument becomes rather complicated because we need to prove properties of the *evaluation procedure* of non-commutative formulas, within the restricted framework of propositional Frege proofs.

The *quasi-polynomial blowup* in Theorem 1.7 depends *solely* on the fact that the reflection principle for non-commutative IPS is efficiently provable (apparently) only when the non-commutative IPS certificate is written as a sum of *homogenous* non-commutative formulas, as we now explain. Note that it is not known whether any arithmetic formula can be turned into a (sum of) homogenous formulas with only a polynomial increase in size (in contrast to the standard efficient homogenization of arithmetic *circuits* by Strassen [33]). Recently Raz [25] showed how to transform an arithmetic formula into (a sum of) homogenous formulas with only a quasi-polynomial increase in size. In Lemma 5.6 we show that:

1. The same construction in [25] holds also for *non-commutative* formulas.
2. This construction for non-commutative formulas can be carried out efficiently inside Frege. That is, if F is a non-commutative formula of size s computing a homogenous non-commutative polynomial over $GF(2)$ and F' is a homogenous non-commutative formula computing the same polynomial with size $s^{O(\log s)}$ (existing by [25]), then Frege admits an $s^{O(\log s)}$ size proof of $F \equiv F'$.

Before we homogenize the non-commutative formulas (according to Raz' construction [25]) we need to *balance* them, so that their depth is logarithmic in their size. We inspect that the recent construction of Hrubeš and Wigderson [11], for balancing non-commutative formulas with division gates (incurring at most a polynomial increase in size) results in a *division-free* formula, when the *initial* non-commutative formula is division-free itself. Therefore, we can assume that the non-commutative IPS certificate is already balanced.

To prove Theorem 1.7 we thus start with a non-commutative IPS certificate π over $GF(2)$ of the polynomial translation of the CNF ϕ, written as a *balanced* non-commutative formula (over $GF(2)$). We then consider this non-commutative polynomial identity *over* $GF(2)$ *as a Boolean tautology* by replacing plus gates with XOR and product gates with AND. We convert this Boolean tautology to a homogenous representation (as described above, using a simulation of [25]). Now, we have a Boolean tautology which we denote by π.

We wish to prove $\neg\phi$ in Frege, using the fact that π is a (massaged version of a) non-commutative IPS certificate. To this end we essentially construct an efficient Frege proof of the correctness of the Raz and Shpilka non-commutative formulas PIT algorithm [26]. The PIT algorithm in [26] uses some basic linear algebraic concepts that might complicate the proof in Frege. However, since we only need to show the *existence* of short Frege proofs for the PIT algorithm's correctness, we can supply *witnesses* to witness the desired linear algebraic objects needed in the proof (these witnesses will be a sequence of linear transformations).

Furthermore, to reason inside Frege directly about the algorithm of [26] is apparently impossible, since this algorithm first converts a non-commutative formula into an *algebraic branching program* (ABP); but apparently the evaluation of ABPs cannot be done with Boolean formulas (and accordingly Frege possibly cannot reason about the evaluation of ABPs). The reason for this apparent inability of Frege to reason about ABP's evaluation is that an ABP is a "sequential" object (an evaluation of an ABP seems to follow from the source to sink, level by level), while Frege operates with formulas, which are "parallel" objects (evaluation of [balanced] formulas can be done in logarithmic time, in case we have enough [separate] processors to perform parallel sub-evaluations of the formula). To overcome this obstacle we show how to perform Raz and Shpilka's PIT algorithm *directly on non-commutative formulas*, without converting the formulas first into ABPs. This technical contribution takes a large part of the argument. We are thus able to prove the following statement, which might be interesting by itself:

▶ **Lemma 1.8.** *If a non-commutative* homogeneous *formula* $F(\overline{x})$ *over* $GF(2)$ *of size* s *is identically zero, then the corresponding Boolean formula* $\neg F_{bool}(\overline{x})$ *(where* F_{bool} *results by replacing* $+$ *with XOR and* $\cdot$ *with AND in* $F(\overline{x})$*) can be proved with a Frege proof of size at most* $s^{O(1)}$.

1.4 Significance and discussion

The propositional-calculus is one of the most natural and central notions in logic, and within proof complexity it has a dominant role as a strong proof system whose structure and complexity is poorly understood. In that respect, our characterization of Frege proofs (and thus propositional-calculus) simply as non-commutative polynomials whose non-commutative formula size corresponds (up to a quasi-polynomial factor) to the size of Frege proofs, should be considered a valuable contribution. Since non-commutative formulas constitute a weak model of computation that is quite well understood, and since the Frege system is considered a strong proof system, and it is not entirely out of the way that Frege – or at least its extension, Extended Frege – is polynomially bounded (i.e., admits polynomial-size proofs for every tautology), our results showing the correspondence between Frege proofs and non-commutative formulas are quite surprising.

This correspondence, between non-commutative formulas and proofs, also gives renewed hope for progress on the major fundamental lower bounds problems in proof complexity: it reduces the problem of proving lower bounds on Frege proofs to the problem of establishing rank lower bounds on matrices associated with non-commutative polynomials, where the non-commutative polynomials are given "semi-explicitly" (that is, they are given in terms of the properties of the non-commutative IPS (Definition 1.2)). It is already known that rank lower bounds yielding strong non-commutative formulas lower bounds are fairly simple (cf. [20]). This then provides a quite compelling evidence that Frege lower bounds, although mostly considered out of reach of current techniques, might not be very far away. Furthermore, our result simplifies greatly the high level-lower bound approach laid out in [10]: the suggested lower bound approach in [10] proposed to move from (commutative) arithmetic circuits lower bounds towards proof complexity lower bounds; but for (commutative) arithmetic circuits there are no known explicit lower bounds, in contrast to non-commutative formulas which constitute a well understood circuit class: both explicit exponential lower bounds and a deterministic PIT algorithms are known for non-commutative formulas.

The new characterization of Frege proofs also sheds light on the correspondence between *circuits and proofs* in proof complexity: in the framework of the ideal proof system, a Frege proof can be seen from the computational perspective as a non-commutative formula.

We also tighten the results of [10]. Namely, by showing that already the non-commutative version of the IPS is sufficient to simulate Frege. As well as by showing *unconditional* efficient simulation of the non-commutative IPS by Frege.

While proving that Frege quasi-polynomially simulates the non-commutative IPS, we demonstrate new simulations of algebraic complexity constructions within proof complexity; these include the homogenization for formulas of Raz [25] and the PIT algorithm for non-commutative formulas by Raz and Shpilka [26]. These proof complexity simulations adds to the known previous such simulations shown in Hrubeš and Tzameret [13] and might be interesting by themselves.

Lastly, this work emphasizes the importance and usefulness of non-commutative models of computation in proof complexity (see [12, 18] for more on this).

1.5 Comparison with previous work

As discussed before, our main characterization of the Frege system is based on a non-commutative version of the IPS system from Grochow and Pitassi [10]. As described above, the non-commutative IPS gives a tighter characterization than the (commutative) IPS in [10]. Thus, our proof system is seemingly weaker than the original (formula version of) IPS, and hence apparently *closer to capture the Frege system.*

Proofs in the original (formula version of the) IPS are arithmetic formulas, and thus any super-polynomial lower bound on IPS refutations implies $\mathsf{VNP}\neq\mathsf{VP}_e$, or in other words, that the permanent does not have polynomial-size arithmetic formulas (Joshua Grochow [personal communication]). This gives a justification of the considerable hardness of proving IPS lower bounds. On the other hand, an exponential-size lower bound on our non-commutative IPS gives only a corresponding lower bound on non-commutative formulas, for which exponential-size lower bounds are already known [20]. Since Frege is quasi-polynomially equivalent to the non-commutative IPS, this means that exponential-size lower bounds on Frege implies merely – at least in the context of the Ideal Proof System – corresponding lower bounds on non-commutative formulas, a result which is however already known. This implies again that there is no strong concrete justification to believe that Frege lower bounds are beyond current techniques.

The work in [35] dealt with propositional proof systems over non-commutative formulas. The difference with the current work is that [35] formulated all proof systems as variants of the polynomial calculus and hence the characterization of a proof system in terms of a *single* non-commutative polynomial is lacking from that work (as well as the consequences we obtained in the current work).

2 Preliminaries

2.1 Frege proof systems

▶ **Definition 2.1** (Boolean formula). Given a set of input variables $x_1, \ldots, x_n$, a *Boolean formula* on the inputs is a rooted tree of fan-in at most two, with edges directed from leaves to the root. Internal nodes are labeled with the Boolean gates $\vee, \wedge, \neg$, and the fan-in of $\vee, \wedge$ is two and the fan-in of $\neg$ is one. The leaves are labeled either with input variables or with $0, 1$ (identified with the truth values false and true, resp.). The entire formula computes the function computed by the gate at the root. Given a formula F, the ***size*** of the formula is the number of Boolean gates in F.

Informally, a Frege proof system is just a standard propositional proof system for proving propositional tautologies (one learns in a basic logic course), having axioms and deduction rules, where proof-lines are written as Boolean formulas. The *size* of a Frege proof is the number of symbols it takes to write down the proof.

The problem of demonstrating super-polynomial size lower bounds on propositional proofs (called also Frege proofs) asks whether there is a family $(F_n)_{n=1}^{\infty}$ of propositional tautological formulas for which there is no polynomial p such that the minimal Frege proof size of F_n is at most $p(|F_n|)$, for all $n \in \mathbb{Z}^+$ (where $|F_n|$ denotes the size of the formula F_n).

▶ **Definition 2.2** (Frege rule). A *Frege rule* is a sequence of propositional formulas $A_0(\overline{x}), \ldots, A_k(\overline{x})$, for $k \leq 0$, written as $\frac{A_1(\overline{x}), \ldots, A_k(\overline{x})}{A_0(\overline{x})}$. In case $k = 0$, the Frege rule is called an *axiom scheme*. A formula F_0 is said to be *derived by the rule* from $F_1, \ldots, F_k$ if $F_0, \ldots, F_k$ are all substitution instances of $A_1, \ldots, A_k$, for some assignment to the $\overline{x}$ variables (that is, there are formulas $B_1, \ldots, B_n$ such that $F_i = A_i(B_1/x_1, \ldots, B_n/x_n)$, for all $i = 0, \ldots, k$). The Frege rule is said to be *sound* if whenever an assignment satisfies the formulas in the upper side $A_1, \ldots, A_k$, then it also satisfies the formula in the lower side A_0.

▶ **Definition 2.3** (Frege proof). Given a set of Frege rules, a *Frege proof* is a sequence of Boolean formulas such that every proof-line is either an axiom or was derived by one of the given Frege rules from previous proof-lines. If the sequence terminates with the Boolean formula A, then the proof is said to be a *proof* of A. The ***size*** of a Frege proof is the the total sizes of all the Boolean formulas in the proof.

A proof system is said to be *implicationally complete* if for all set of formulas T, if T semantically implies F, then there is a proof of F using (possibly) axioms from T. A proof system is said to be sound if it admits proofs of only tautologies (when not using auxiliary axioms, like in the T above).

▶ **Definition 2.4** (Frege proof system). Given a propositional language and a set P of sound Frege rules, we say that P is a *Frege proof system* if P is implicationally complete.

Note that a Frege proof is always sound since the Frege rules are assumed to be sound. We do not need to work with a specific Frege proof system, since a basic result in proof complexity states that every two Frege proof systems, even over different languages, are polynomially equivalent [29].

2.2 Algebraic proof systems

In this section, we give the definitions the algebraic proof systems Polynomial Calculus over Formulas ($\mathcal{F}$-$\mathcal{PC}$) defined by Grigoriev and Hirsch [9]. We start with the definition of a non-commutative formula:

▶ **Definition 2.5** (Non-commutative formula). Let $\mathbb{F}$ be a field and $x_1, x_2, \ldots$ be variables. A *noncommutative arithmetic formula* (or *noncommutative formula* for short) is a labeled tree, with edges directed from the leaves to the root, and with fan-in at most two, such that there is an order on the edges coming into a node (the first edge is called the *left* edge and the second one the *right* edge). Every leaf of the tree (namely, a node of fan-in zero) is labeled either with an input variable x_i or a field $\mathbb{F}$ element. Every other node of the tree is labeled either with $+$ or $\times$ (in the first case the node is a plus gate and in the second case a product gate). We assume that there is only one node of out-degree zero, called *the root.* A noncommutative formula *computes* a noncommutative polynomial in $\mathbb{F}\langle x_1, \ldots, x_n\rangle$ in the following way. A leaf computes the input variable or field element that labels it. A plus gate computes the sum of polynomials computed by its incoming nodes. A product gate computes the *noncommutative* product of the polynomials computed by its incoming nodes according to the order of the edges. (Subtraction is obtained using the constant -1.) The output of the formula is the polynomial computed by the root. The *depth* of a formula is the maximal length of a path from the root to the leaf. The ***size*** of a noncommutative formula f is the total number of nodes in its underlying tree, and is denoted $|f|$.

The definition of (a commutative) arithmetic formula is almost identical:

▶ **Definition 2.6** (Arithmetic formula). An *arithmetic formula* is defined in a similar way to a noncommutative formula, except that we ignore the order of multiplication (that is, a product node does not have order on its children and there is no order on multiplication when defining the polynomial computed by a formula).

Given a pair of non-commutative formulas F and G and a variable x_i, we denote by $F[G/x_i]$ the formula F in which every occurrence of x_i is substituted by the formula G.

Note that an arithmetic formula is a syntactic object. For example, $x_1 + x_2$ and $x_2 + x_1$ are different formulas because commutativity might not hold (even if commutativity holds, we will regard them as different formulas. And in the proof system $\mathcal{F}$-$\mathcal{PC}$ they can be *derived* from each other via the "commutativity of addition").

2.2.1 Polynomial calculus over formulas F-PC system

The $\mathcal{F}$-$\mathcal{PC}$ proof system defined by Grigoriev and Hirsch [9] operates with arithmetic formulas (as purely syntactic terms).

▶ **Definition 2.7** ($\mathcal{F}$-$\mathcal{PC}$ [9]). Fix a field $\mathbb{F}$. Let $F := \{f_1, \ldots, f_m\}$ be a collection of *formulas* computing polynomials from $\mathbb{F}[x_1, \ldots, x_n]$ (note here that we are talking about formulas (treated as syntactic terms), and *not* polynomials. Also notice that all formulas in $\mathcal{F}$-$\mathcal{PC}$ are (commutative) formulas computing (commutative) polynomials). Let the set of axioms be the following formulas:

Boolean axioms $\quad x_i \cdot (1 - x_i)\,, \qquad$ for all $1 \le i \le n\,.$

A sequence $\pi = (\Phi_1, \ldots, \Phi_\ell)$ of formulas computing polynomials from $\mathbb{F}[x_1, \ldots, x_n]$ is said to be *an $\mathcal{F}$-$\mathcal{PC}$ proof of Φ_ℓ from F*, if for every $i \in [\ell]$ we have one of the following:

1. $\Phi_i = f_j$, for some $j \in [m]$;
2. Φ_i is a Boolean axiom;
3. Φ_i was deduced by one of the following inference rules from previous proof-lines Φ_j, Φ_k , for $j, k < i$:

Product

$$\frac{\Phi}{x_r \cdot \Phi}\,, \qquad \text{for } r \in [n]\,.$$

Addition

$$\frac{\Phi \qquad \Theta}{a \cdot \Phi + b \cdot \Theta}\,, \qquad \text{for } a, b \in \mathbb{F}\,.$$

(Where $\Phi, x_r \cdot \Phi, \Theta, a \cdot \Phi, b \cdot \Theta$ are *formulas* constructed as displayed; e.g., $x_r \cdot \Phi$ is the formula with product gate at the root having the formulas x_r and Φ as children.)(In [9] the product rule of $\mathcal{F}$-$\mathcal{PC}$ is defined so that one can derive $\Theta \cdot \Phi$ from Φ, where Θ is any formula, and not just a variable. However, the definition of $\mathcal{F}$-$\mathcal{PC}$ in [9] and our Definition 2.7 polynomially-simulate each other.)

4. Φ_i was deduced from previous proof-line Φ_j, for $j < i$, by one of the following *rewriting rules* expressing the polynomial-ring axioms (where f, g, h range over all arithmetic formulas computing polynomials in $\mathbb{F}[x_1, \ldots, x_n]$):
   Zero rule $0 \cdot f \leftrightarrow 0$
   Unit rule $1 \cdot f \leftrightarrow f$
   Scalar rule $t \leftrightarrow \alpha$, where t is a formula containing no variables (only field $\mathbb{F}$ elements) that computes the constant $\alpha \in \mathbb{F}$.
   Commutativity rules $f + g \leftrightarrow g + f\,, \qquad f \cdot g \leftrightarrow g \cdot f$
   Associativity rule $f + (g + h) \leftrightarrow (f + g) + h\,, \qquad f \cdot (g \cdot h) \leftrightarrow (f \cdot g) \cdot h$
   Distributivity rule $f \cdot (g + h) \leftrightarrow (f \cdot g) + (f \cdot h)$

(The semantics of an $\mathcal{F}$-$\mathcal{PC}$ proof-line p_i is the polynomial equation $p_i = 0$.) An $\mathcal{F}$-$\mathcal{PC}$ *refutation of* F is a proof of the formula 1 from F. The ***size*** of an $\mathcal{F}$-$\mathcal{PC}$ proof π is defined as the total size of all formulas in π and is denoted by $|\pi|$.

▶ **Definition 2.8** (Tree-like $\mathcal{F}$-$\mathcal{PC}$)**.** A system $\mathcal{F}$-$\mathcal{PC}$ is a *tree-like* $\mathcal{F}$-$\mathcal{PC}$ if every derived arithmetic formula in the proof system is used only once (and if it is needed again, it must be derived once more).

2.2.1.1 Translation of Boolean formulas into polynomial equations

The proof system $\mathcal{F}$-$\mathcal{PC}$ can be considered as a propositional proof system for Boolean tautologies (namely, Boolean formulas that are true under any assignment). Given a Boolean formula T in the propositional variables $x_1, \ldots, x_n$ we can transform T into a set of polynomial equations by encoding it into a set of arithmetic formulas where each clause in the CNF corresponds to an arithmetic formula by replacing $\wedge$ with $\times$, $\vee$ with $+$ and $\neg x$ with $1 - x$; and for each variable x_i, add $x_i^2 - x_i$ (called the *Boolean axioms*) to guarantee that every satisfying assignment to the variables is a 0-1 assignment. Then the given CNF is a tautology if and only if the set of arithmetic formulas have no common root.

▶ **Definition 2.9** (Polynomially Simulation)**.** Let $\mathcal{P}_1, \mathcal{P}_2$ be two proof systems for the same language L (in case the proof systems are for two different languages we fix a translation from one language to the other, as described above). We say that $\mathcal{P}_2$ *polynomially simulates* $\mathcal{P}_1$ if given a $\mathcal{P}_1$ proof (or refutation) π of a F, then there exists a proof (respectively, refutation) of F in $\mathcal{P}_2$ of size polynomial in the size of π. In case $\mathcal{P}_2$ polynomially simulates $\mathcal{P}_1$ while $\mathcal{P}_1$ does not polynomially simulates $\mathcal{P}_2$ we say that $\mathcal{P}_2$ is *strictly stronger* than $\mathcal{P}_1$.

In [9], it was shown that $\mathcal{F}$-$\mathcal{PC}$, as well as tree-like $\mathcal{F}$-$\mathcal{PC}$, polynomially simulate Frege. We repeat the argument for the convenience of the reader:

▶ **Theorem 2.10** ([9])**.** *Tree-like $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates Frege.*

Proof. The following was shown in [9]:

▶ **Theorem 2.11** (Theorem 3, [9]). *The system $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates the Frege system.*

Moreover, inspecting the proof of the above theorem, we can observe that tree-like Frege proofs are simulated by tree-like $\mathcal{F}$-$\mathcal{PC}$ proofs:

▶ **Lemma 2.12.** *Tree-like $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates tree-like Frege systems.*

But Krajíček showed that tree-like Frege and Frege are polynomially equivalent:

▶ **Theorem 2.13** ([14]). *Tree-like Frege proofs polynomially simulate Frege proofs.*

Thus, by this theorem and by Lemma 2.11, tree-like $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates the Frege system. ◀

3 The non-commutative ideal proof system

The non-commutative ideal proof system (non-commutative IPS for short) is an algebraic refutation system in which a refutation is a single non-commutative polynomial. In the next section we show that when the non-commutative IPS refutations are written as *non-commutative formulas* then the non-commutative IPS polynomially simulates tree-like $\mathcal{F}$-$\mathcal{PC}$, and hence polynomially simulates the Frege proof system (by [9]).

▶ **Definition 3.1** (Non-commutative IPS). Let $\mathbb{F}$ be a field. Assume that $F_1(\overline{x}) = F_2(\overline{x}) = \cdots = F_m(\overline{x}) = 0$ is a system of non-commutative polynomial equations from $\mathbb{F}\langle x_1, \ldots, x_n\rangle$, and suppose that the following set of equations (axioms) are included in the $F_i(\overline{x})$'s:

Boolean axiom: $x_i \cdot (1 - x_i)\,, \quad$ for all $1 \le i \le n\,;$

Commutator axiom: $x_i \cdot x_j - x_j \cdot x_i\,, \quad$ for all $1 \le i < j \le n\,.$

Suppose that the $F_i(\overline{x})$'s have no common 0-1 solutions. (One can check that the $F_i(\overline{x})$'s have no common 0-1 solutions in $\mathbb{F}$ iff they do not have a common 0-1 solution in every $\mathbb{F}$-algebra.) A *non-commutative IPS refutation* (or *certificate*) that the system of $F_i(\overline{x})$'s is unsatisfiable is a ***non-commutative polynomial*** $\mathfrak{F}(\overline{x}, \overline{y})$ in the variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ (i.e. $\mathfrak{F} \in \mathbb{F}\langle \overline{x}, \overline{y}\rangle$), such that:

1. $\mathfrak{F}(x_1, \ldots, x_n, \overline{0}) = 0$; and
2. $\mathfrak{F}(x_1, \ldots, x_n, F_1(\overline{x}), \ldots, F_m(\overline{x})) = 1$.

In this paper we assume that the non-commutative IPS refutation is written as a *non-commutative formula*. Hence the ***size*** of a non-commutative IPS refutation is the minimal size of a non-commutative formula computing the non-commutative IPS refutation.

▶ **Comment.**
1. The identities in items 1 and 2 in Definition 3.1 are *formal* identities of polynomials (i.e., in 1 the polynomial in the left hand side has a zero coefficient for every monomial, and in 2 the only nonzero monomial is the monomial 1).
2. In order to prove that a system of *commutative* polynomial equations $\{P_i = 0\}$ (where each P_i is expressed as an arithmetic formula) has no common roots in non-commutative IPS, we write each P_i *as a non-commutative formula* (in some way; note that there is no unique way to do this).
3. When we write $P \cdot Q - Q \cdot P$ where P, Q are formulas (e.g., x_i and x_j, resp.), we mean $((P \cdot Q) + (-1 \cdot (Q \cdot P)))$.

4 Non-commutative ideal proof system polynomially simulates Frege

Here we show that the non-commutative IPS polynomially simulates Frege.

▶ **Theorem 4.1** (restatement of Theorem 1.4). *The non-commutative IPS refutation system, when refutations are written as non-commutative formulas, polynomially simulates Frege systems. More precisely, for every propositional tautology T, if T has a polynomial-size Frege proof then there is a non-commutative IPS certificate (with integer coefficients) of polynomial non-commutative formula size.*

Recall that Raz and Shpilka [26] gave a deterministic polynomial-time PIT algorithm for non-commutative formulas (over any field):

▶ **Theorem 4.2** (PIT for non-commutative formulas [26]). *There is a deterministic polynomial-time algorithm that decides whether a given noncommutative formula over a field $\mathbb{F}$ computes the zero polynomial 0. (We assume here that the field $\mathbb{F}$ can be efficiently represented (e.g., the field of rationals).)*

Now, since we write refutations as non-commutative formulas we can use the theorem above to check in *deterministic* polynomial-time the correctness of non-commutative IPS refutations, obtaining:

▶ **Corollary 4.3** (restatement of Corollary 1.5). *The non-commutative IPS is a sound and complete Cook-Reckhow refutation system. That is, it is a sound and complete refutation system for unsatisfiable propositional formulas in which refutations can be checked for correctness in deterministic polynomial-time.*

To prove Theorem 4.1, we will show in Section 4.1 that the non-commutative IPS polynomially-simulates tree-like $\mathcal{F}$-$\mathcal{PC}$ (Definition 2.7), which suffices to complete the proof due to Theorem 2.10.

4.1 Non-commutative IPS polynomially simulates tree-like F-PC

For convenience, let $C_{i,j}$ denote the commutator axiom $x_i \cdot x_j - x_j \cdot x_i$, for $i, j \in [n], i \neq j$.

▶ **Theorem 4.4.** *Non-commutative IPS polynomially simulates Tree-like $\mathcal{F}$-$\mathcal{PC}$ (Definition 2.7).*

Proof. Let $F_1, \ldots, F_m$ be arithmetic formulas over the variables $x_1, \ldots, x_n$. Note that an arithmetic formula is a syntactic term in which the children of gates are ordered. We thus can treat a (commutative) arithmetic formula as a *non-commutative* arithmetic formula by taking the *order* on the children of products gates to be the order of non-commutative multiplication.

Suppose $\mathcal{F}$-$\mathcal{PC}$ has a poly(n)-size tree-like refutation $\pi := (L_1, \ldots, L_k)$ of the F_i's (i.e., a proof of the polynomial 1 from $F_1, \ldots, F_m$), where each L_j is an arithmetic formula. We construct a corresponding non-commutative IPS refutation of the F_i's from this $\mathcal{F}$-$\mathcal{PC}$ tree-like refutation. Denote by $|\pi|$ the size of π. We have the following:

▶ **Lemma 4.5.** *For each $i \in [k]$, there exists a non-commutative formula ϕ_i such that:*

1. $\phi_i(\overline{x}, \overline{0}) = 0$;
2. $\phi_i(\overline{x}, F_t, C_{j,j'}) = L_i$, *where $t \in [m]$, $j, j' \in [n]$, $j < j'$; (this is an abuse of notation meaning $\phi_i(\overline{x}, F_1, \ldots, F_m, C_{1,2}, C_{1,3} \ldots, C_{n-1,n})$. We use a similar abuse of notation in the sequel.)*

3. $|\phi_i| \le \left(\sum_{\ell\in A_i}|L_\ell|\right)^4$, *where* $A_i \subset [k]$ *refers to the indices of the* $\mathcal{F}$-$\mathcal{PC}$ *proof-lines involved in deriving* L_i. *(For example, if* L_i *is derived by* L_α *and* L_α *is derived by* L_β *for some* $\beta < \alpha < i \in [k]$, *then we say that* α, β *are both involved for deriving* L_i.*)*

Note that if the lemma holds, then ϕ_k will be a non-commutative IPS proof because it has the property that $\phi_k(\overline{x}, \overline{0}) = 0$ and $\phi_k(\overline{x}, F_t, C_{j,j'}) = L_k = 1$, where $t \in [m], j, j' \in [n], j \neq j'$. And its size is bounded by $\left(\sum_{\ell\in A_k}|L_\ell|\right)^4 \le \left(\sum_{\ell\in[k]}|L_\ell|\right)^4 \le O(|\pi|^4)$. Thus, non-commutative IPS polynomially simulates tree-like $\mathcal{F}$-$\mathcal{PC}$. ◀

We construct ϕ_i by induction on the length k of the refutation π. That is, for i from 1 to k, we construct the non-commutative formula $\phi_i(\overline{x}, \overline{y})$ according to L_i, as follows:

Case 1:

The L_i is the input axiom F_j for some $j \in [m]$.

Let $\phi_i := y_j$. Obviously, $\phi_i(\overline{x}, 0) = 0, \phi_i(\overline{x}, F_t, C_{\alpha,\beta}) = F_j = L_i$ and $|\phi_i| = 1 \le |L_i|^4$.

Case 2:

The L_i is derived from an inference rule from previous proof-lines $L_j, L_{j'}$, for $j, j' < i$. Then we divide this case into two parts.

Part (1): The L_i is derived from the addition rule $L_i = aL_j + bL_{j'}$. Put $\phi_i := a\phi_j + b\phi_{j'}$ where $a, b \in \mathbb{F}$. Thus, $\phi_i(\overline{x}, 0) = a\phi_j(\overline{x}, 0) + b\phi_{j'}(\overline{x}, 0) = 0$, $\phi_i(\overline{x}, F_t, C_{\alpha,\beta}) = aL_j + bL_{j'} = L_i$ and $|\phi_i| = |\phi_j| + |\phi_{j'}| + 3 \le \left(\sum_{\ell\in A_j}|L_\ell|\right)^4 + \left(\sum_{\ell\in A_{j'}}|L_\ell|\right)^4 + 3 \le \left(\sum_{\ell\in A_i}|L_\ell|\right)^4$ (where the right most inequality holds since π is a *tree-like* refutation and hence $A_j \cap A_{j'} = \emptyset$).
Part (2): The L_i is derived from the product rule $L_i = x_r \cdot L_{j'}$ for $r \in [n]$. Put $\phi_i := (x_r \cdot \phi_j)$. Then $\phi_i(\overline{x}, 0) = x_r \cdot \phi_j(\overline{x}, 0) = 0, \phi_i(\overline{x}, F_t, C_{\alpha,\beta}) = x_r \cdot L_j = L_i$ and $|\phi_i| = |\phi_j| + 2 \le \left(\sum_{\ell\in A_j}|L_\ell|\right)^4 + 2 \le \left(\sum_{\ell\in A_i}|L_\ell|\right)^4$.

Case 3:

The L_i is derived from L_j by a rewriting rule excluding the commutative rule of multiplication. Let $\phi_i := \phi_j$. The non-commutative ϕ_i satisfies the properties claimed trivially since all the rewriting rules (excluding the commutative rule of multiplication) express the non-commutative polynomial-ring axioms, and thus cannot change the polynomial computed by a non-commutative formula. And $|\phi_i| = |\phi_j| \le \left(\sum_{\ell\in A_i}|L_\ell|\right)^4$.

Case 4:

The L_i is derived from L_j by a single application of the commutative rule of multiplication. Then by Lemma 4.6 below, we can construct a non-commutative formula ϕ_{L_i,L_j} such that $\phi_i := (\phi_j + \phi_{L_i,L_j})$ satisfies the desired properties (stated in Lemma 4.5).

▶ **Lemma 4.6.** *Let* L_i, L_j *be non-commutative formulas such that* L_i *can be derived from* L_j *via the commutative rule of multiplication. Then there is a non-commutative formula* $\phi_{L_i,L_j}(\overline{x}, \overline{y})$ *in variables* $\{x_\ell, y_{\alpha,\beta},\ \ell \in [n], \alpha < \beta \in [n]\}$, *such that:*
1. $\phi_{L_i,L_j}(\overline{x}, \overline{0}) = 0$;
2. $\phi_{L_i,L_j}(\overline{x}, C_{\alpha,\beta}) = L_i - L_j$;
3. $\left|\phi_{L_i,L_j}\right| \le |L_i|^2 \, |L_j|^2$.

Proof. We define the non-commutative formula ϕ_{L_i,L_j} inductively as follows:

- If $L_i = (P \cdot Q)$, and $L_j = (Q \cdot P)$, then ϕ_{L_i,L_j} is defined to be the formula constructed in Lemma 4.7 below.
- If $L_i = (P \cdot Q)$, $L_j = (P' \cdot Q')$.
  Case 1. If $P = P'$, then let $\phi_{L_i,L_j} := (P \cdot \phi_{Q,Q'})$.
  Case 2. If $Q = Q'$, then let $\phi_{L_i,L_j} := (\phi_{P,P'} \cdot Q)$.
- If $L_i = (P + Q)$, $L_j = (P' + Q')$.
  Case 1. If $P = P'$, then let $\phi_{L_i,L_j} = \phi_{Q,Q'}$.
  Case 2. If $Q = Q'$, then let $\phi_{L_i,L_j} = \phi_{P,P'}$.

By induction, one could check the construction satisfies the desired properties. ◀

▶ **Lemma 4.7.** *For any pair P, Q of two non-commutative formulas there exists a non-commutative formula F in variables $\{x_\ell, y_{i,j},\ \ell \in [n], i < j \in [n]\}$ such that:*

1. $F(\overline{x}, \overline{0}) = 0$;
2. $F(\overline{x}, C_{i,j}) = P \cdot Q - Q \cdot P$;
3. $|F| = |P|^2 |Q|^2$. *e*

Proof. Let $s(P, Q)$ denote the smallest size of F satisfying the above properties. We will show that $s(P, Q) \le |P|^2 \cdot |Q|^2$ by induction on $\max(|P|, |Q|)$.

Base case: $|P| = |Q| = 1$.
In this case both P and Q are constants or variables, thus $s(P, Q) = 1 \le |P|^2 |Q|^2$.

In the following induction step, we consider the case that $|P| \ge |Q|$ (which is symmetric for the case $|P| < |Q|$).

Induction step: Assume that $|P| \ge |Q|$ (the case $|P| < |Q|$ is similar).
Case 1: The root of P is addition.
Let $P = (P_1 + P_2)$. We have (after rearranging):

$$P \cdot Q - Q \cdot P = ((P_1 \cdot Q - Q \cdot P_1) + (P_2 \cdot Q - Q \cdot P_2))$$

By induction hypothesis, we have $s(P, Q) \le s(P_1, Q) + 1 + s(P_2, Q) \le |P_1|^2 |Q|^2 + 1 + |P_2|^2 |Q|^2 \le (|P_1| + |P_2| + 1)^2 |Q|^2 = |P|^2 \cdot |Q|^2$.
Case 2: The root of P is a product gate.
Let $P = (P_1 \cdot P_2)$. By rearranging:

$$P \cdot Q - Q \cdot P = ((P_1 \cdot (P_2 \cdot Q - Q \cdot P_2)) + ((P_1 \cdot Q - Q \cdot P_1) \cdot P_2))$$

By induction hypothesis, we have $s(P, Q) = |P_1| + 1 + s(P_2, Q) + 1 + s(P_1, Q) + 1 + |P_2| \le |P_1| + 1 + |P_2|^2 |Q|^2 + 1 + |P_1|^2 |Q|^2 + 1 + |P_2| \le (|P_1| + |P_2| + 1)^2 |Q|^2 = |P|^2 \cdot |Q|^2$. ◀

5 Frege quasi-polynomially simulates non-commutative IPS

In this section we prove that the Frege system quasi-polynomially simulates the non-commutative IPS (over $GF(2)$). Together with Theorem 4.4, this gives a new characterization (up to a quasi-polynomial increase in size) of propositional Frege proofs as non-commutative arithmetic formulas.

We use the notation in Section 1.3.3: for a clause k_i in a CNF $\phi = k_1 \wedge \ldots \wedge k_m$, we denote by Q_i^ϕ the non-commutative formula translation $\mathrm{tr}'(k_i)$ of the clause k_i (Definition 1.6). Thus, $\neg x$ is translated to x, x is translated to $1 - x$ and $f_1 \cdots f_r$ is translated to $\prod_i \mathrm{tr}'(f_i)$

(considered as a tree of product gates with $\mathrm{tr}'(f_i)$ as leaves), and where the formulas are over $GF(2)$ (meaning that $1-x$ is in fact $1+x$).

▶ **Theorem 5.1** (Main quasi-polynomial simulation). *For a 3CNF $\phi = k_1 \wedge \ldots \wedge k_m$ where $Q_1^\phi, \ldots, Q_m^\phi$ are the corresponding polynomial equations for the clauses, if there is a non-commutative IPS refutation of size s of $Q_1^\phi, \ldots, Q_m^\phi$ over $GF(2)$, then there is a Frege proof of size $s^{O(\log s)}$ of $\neg\phi$.*

The rest of the section is dedicated to proving Theorem 5.1. Due to lack of space we refer the reader to the full version of this work [19] for complete proofs. Here we shall only outline the main parts of the proof (see also Section 1.3.3).

5.1 Balancing non-commutative formulas

First we show that a non-commutative formula of size s can be balanced to an equivalent formula of depth $O(\log s)$, and thus we can assume that the non-commutative IPS certificate is already given as a balanced formula (this is needed for what follows). Both the statement of the balancing construction and its proof are similar to Proposition 4.1 in Hrubeš and Wigderson [11] (which in turn is similar to the case of commutative formulas with division gates in Brent [5]). Note that a formula of a logarithmic depth must have a polynomial-size. (Thus, in what follows, without loss of generality we will assume that F is given already in a balanced form, namely has depth $O(\log s)$ and polynomial-size which, for simplicity, we denoted as s.)

▶ **Lemma 5.2.** *Assume that a non-commutative polynomial p can be computed by a formula of size s. Then p can be computed by a formula of depth $O(\log s)$ (and hence of polynomial-size).*

5.2 The reflection principle

Here we show that the existence of a non-commutative IPS refutation of size s and depth $O(\log s)$, implies the existence of a Frege proof with size $s^{O(\log s)}$ of $\neg\phi$. This is done by proving a *reflection principle* for the non-commutative IPS system in Frege. As mentioned in the introduction, informally, a reflection principle for a given proof system P is a statement that says that if a formula is provable in P then the formula is also *true*. Thus, suppose we have a short Frege-proof of the reflection principle for P, having the form:

$$\text{“}([\pi] \text{ is a } P\text{-proof of } [T]) \longrightarrow T\text{”},$$

where $[T]$ and $[\pi]$ are some reasonable encodings of the tautology T and its P-proof π, respectively. Then, we can easily obtain a Frege proof of T assuming we have a P-proof of T.

Let F be a non-commutative formula over $GF(2)$ and let $\overline{Q}^\phi(\overline{x})$ denote the vector $(Q_1^\phi, \ldots, Q_m^\phi)$ (see Theorem 5.1). First, note that F is a non-commutative IPS proof of ϕ only if it has the following two properties:

$$F\left(\overline{x}, \overline{0}\right) = 0, \qquad F\left(\overline{x}, \overline{Q}^\phi(\overline{x})\right) = 1, \tag{1}$$

showing the unsatisfiability of $\overline{Q}^\phi(\overline{x}) = 0$, and hence showing $\neg\phi$ is a tautology. We can treat F as a Boolean formula, as follows:

▶ **Definition 5.3** (Booleanization F_{bool}). Let $F(\overline{x})$ be a non-commutative formula over $GF(2)$ in the (algebraic) variables $\overline{x}$. We denote by $F_{bool}(\overline{p})$ the Boolean formula in the (propositional) variables $\overline{p}$ obtained by turning every plus gate and multiplication gate to $\oplus$ (i.e., XOR) and $\wedge$ gates, respectively, and turning the input variables $\overline{x}$ into the propositional variables $\overline{p}$. We sometimes write F and F_{bool} without explicitly mentioning the $\overline{x}$ and $\overline{p}$ variables, respectively.

When we consider $F = F(\overline{x}, \overline{y})$ (with both the $\overline{x}$ and $\overline{y}$ variables), F_{bool} denotes the Booleanization of F when the variables $\overline{x}$ are replaced by $\overline{p}$ and the variables $\overline{y}$ are still written as $\overline{y}$. Note that for any 0-1 assignment, F and F_{bool} have the same value. Therefore, by the properties in (1), we know:

$$\neg F_{bool}\left(\overline{p}, \overline{0}\right), \qquad F_{bool}\left(\overline{p}, \overline{Q}^{\phi}_{bool}(\overline{p})\right) \tag{2}$$

are *both tautologies* (though we still need to proof that their Frege proofs are short).

To conclude Theorem 5.1, we first prove in Frege $\neg\phi$ based on (2) (this is done in Lemma 5.4 below which is not hard to establish), and then we show that there exists an $s^{O(\log s)}$ Frege proof of (2) (which is done in Lemma 5.5 in the next section, and requires much more work).

▶ **Lemma 5.4.** $\left(\left(\neg F_{bool}\left(\overline{p}, \overline{0}\right)\right) \wedge F_{bool}\left(\overline{p}, \overline{Q}^{\phi}_{bool}(\overline{p})\right)\right) \to \neg\phi$ *can be proved with a polynomial-size Frege proof.*

Having this lemma, it remains to show a quasi-polynomial-size proof of (2). We denote $\neg F_{bool}\left(\overline{p}, \overline{0}\right)$ and $\neg\left(1 \oplus F_{bool}\left(\overline{p}, \overline{Q}^{\phi}_{bool}(\overline{p})\right)\right)$ by

$$F'_{bool}(\overline{p}), \quad F''_{bool}(\overline{p}), \quad \text{respectively.} \tag{3}$$

Note that the substitutions of the constants 0 or the constant depth formulae $\overline{Q}^{\phi}_{bool}$ in F cannot increase the depth of F too much (i.e., can add at most a constant to the size of F). In other words, the depths of the formulae in (3) are still $O(\log s)$.

5.3 Non-commutative formula identities have quasi-polynomial-size proofs

Recall that a (commutative or non-commutative) multivariate polynomial f is *homogeneous* if every monomial in f has the same total degree. For each $0 \le j \le d$, denote by $f^{(j)}$ the homogenous part of degree j of f, that is, the sum of all monomials (together with their coefficient from the field) in f of total degree j. We say that a formula is *homogeneous* if each of its gates computes a *homogeneous* polynomial (see Definition 2.5 for the definition of a polynomial computed by a gate or a formula).

To complete the proof of Theorem 5.1 it remains to prove the following:

▶ **Lemma 5.5.** *If a non-commutative formula $F(\overline{x})$ with 0-1 coefficients of size s and depth $O(\log s)$ is identically zero, then the corresponding Boolean formula $\neg F_{bool}(\overline{p})$ admits a Frege proof of size $s^{O(\log s)}$.*

5.4 Homogenization of non-commutative formulas has short Frege proofs

To complete the proof of Lemma 5.5 it remains to prove Lemmas 5.6 and 1.8 in what follows. Lemma 5.6 states that Raz' construction from [25] for homogenizing arithmetic formulas is efficiently provable in Frege (and is also applicable to non-commutative formulas):

▶ **Lemma 5.6.** *If F is a non-commutative formula of size s and depth $O(\log s)$ and $F^{(0)}, \ldots, F^{(s)}$ are the homogenous formulae computing F's homogenous parts of degrees $0, \ldots, s$, respectively, constructed according to [25], then there exists an $s^{O(\log s)}$-size Frege proof of:*

$$\bigoplus_{i=0}^{s+1} F^{(i)} \leftrightarrow F_{bool}.$$

5.5 Homogenous non-commutative formula identities have polynomial-size Frege proofs

To conclude Theorem 5.1 it remains to prove Lemma 5.7 below, which is the main technical lemma of the whole argument. It states that a non-commutative syntactic-homogenous formula identity over $GF(2)$ has polynomial-size Frege proofs (considered as a Boolean tautology). The proof of this lemma is somewhat lengthy as it entails us to show that the Raz and Shpilka polynomial-time PIT algorithm for non-commutative formulas can be "simulated" efficiently with Frege proofs. Here we just state formally Lemma 1.8 and refer the reader to the full version of the paper [19] for a complete proof of this lemma.

▶ **Lemma 5.7** (Main technical lemma). *There exists a constant c such that if a non-commutative syntactic homogeneous formula $F(\overline{x})$ over $GF(2)$ of size s is identically zero, then the corresponding Boolean tautology $\neg F_{bool}(\overline{p})$ can be proved with a Frege proof of size at most s^c (for sufficiently large s).*

Acknowledgements. We thank Joshua Grochow for helpful comments.

References

1 Miklós Ajtai. The complexity of the pigeonhole principle. In *Proceedings of the IEEE 29th Annual Symposium on Foundations of Computer Science*, pages 346–355, 1988.

2 Albert Atserias, Nicola Galesi, and Pavel Pudlák. Monotone simulations of non-monotone proofs. *J. Comput. System Sci.*, 65(4):626–638, 2002. Special issue on complexity, 2001 (Chicago, IL).

3 Albert Atserias, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint propagation as a proof system. In *CP*, pages 77–91, 2004.

4 Maria Luisa Bonet, Samuel R. Buss, and Toniann Pitassi. Are there hard examples for Frege systems? In *Feasible mathematics, II (Ithaca, NY, 1992)*, volume 13 of *Progr. Comput. Sci. Appl. Logic*, pages 30–56. Birkhäuser Boston, Boston, MA, 1995.

5 Richard P. Brent. The parallel evaluation of general arithmetic expressions. *J. ACM*, 21(2):201–206, 1974.

6 Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 174–183, New York, 1996. ACM.

7 Stephen Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity.* ASL Perspectives in Logic. Cambridge University Press, 2010.

8 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

9 Dima Grigoriev and Edward A. Hirsch. Algebraic proof systems over formulas. *Theoret. Comput. Sci.*, 303(1):83–102, 2003. Logic and complexity in computer science (Créteil, 2001).

10 Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing. In *55th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2014. Also available as arXiv:1404.3820 [cs.CC].

11 Pavel Hrubeš and Avi Wigderson. Non-commutative arithmetic circuits with division. In *Innovations in Theoretical Computer Science, ITCS'14, Princeton, NJ, USA, January 12-14, 2014*, pages 49–66, 2014.

12 Pavel Hrubeš. How much commutativity is needed to prove polynomial identities? *Electronic Colloquium on Computational Complexity, ECCC*, (Report no.: TR11-088), June 2011.

13 Pavel Hrubeš and Iddo Tzameret. Short proofs for the determinant identities. In *Proceedings of the 44th Annual ACM Symposium on the Theory of Computing (STOC)*, New York, 2012. ACM.

14 Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*, volume 60 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1995.

15 Jan Krajíček. An exponential lower bound for a constraint propagation proof system based on ordered binary decision diagrams. *J. Symbolic Logic*, 73(1):227–237, 2008.

16 Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Structures Algorithms*, 7(1):15–39, 1995.

17 Jan Krajíček. *Forcing with random variables and proof complexity*. London Mathematical Society Lecture Note Series, No.382. Cambridge University Press, 2011.

18 Fu Li and Iddo Tzameret. Generating matrix identities and proof complexity. *Electronic Colloquium on Computational Complexity, TR13-185*, 2013. arXiv:1312.6242 [cs.CC] `http://arxiv.org/abs/1312.6242`.

19 Fu Li, Iddo Tzameret, and Zhengyu Wang. Non-commutative formulas and Frege lower bounds: a new characterization of propositional proofs. 2014. arXiv:/1412.8746 [cs.CC] `http://arxiv.org/abs/1412.8746`.

20 N. Nisan. Lower bounds for non-commutative computation. *Proceedings of the 23th Annual ACM Symposium on the Theory of Computing*, pages 410–418, 1991.

21 Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Comput. Complexity*, 3(2):97–140, 1993.

22 Pavel Pudlák. On the complexity of the propositional calculus. In *Sets and proofs (Leeds, 1997)*, volume 258 of *London Math. Soc. Lecture Note Ser.*, pages 197–218. Cambridge Univ. Press, Cambridge, 1999.

23 Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing, Vol. 2, article 6*, 2006.

24 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2), 2009.

25 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40, 2013.

26 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non commutative models. *Computational Complexity*, 14(1):1–19, 2005.

27 Ran Raz and Iddo Tzameret. Resolution over linear equations and multilinear proofs. *Ann. Pure Appl. Logic*, 155(3):194–224, 2008.

28 Ran Raz and Iddo Tzameret. The strength of multilinear proofs. *Computational Complexity*, 17(3):407–457, 2008.

29 Robert Reckhow. *On the lengths of proofs in the propositional calculus*. PhD thesis, University of Toronto, 1976. Technical Report No . 87.

30 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

31 Nathan Segerlind. Nearly-exponential size lower bounds for symbolic quantifier elimination algorithms and OBDD-based proofs of unsatisfiability. *Electronic Colloquium on Computational Complexity*, January 2007. ECCC, TR07-009.

32 Philip M. Spira. On time-hardware complexity tradeoffs for boolean functions. In *Fourth International Symposium on Systems Sciences*, pages 525–527, 1971.

33 Volker Strassen. Vermeidung von divisionen. *J. Reine Angew. Math.*, 264:182–202, 1973. (in German).

34 Iddo Tzameret. *Studies in Algebraic and Propositional Proof Complexity.* PhD thesis, Tel Aviv University, 2008.

35 Iddo Tzameret. Algebraic proofs over noncommutative formulas. *Information and Computation*, 209(10):1269–1292, 2011.

36 Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on the Theory of Computing*, pages 249–261. ACM, 1979.

37 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 216–226. Springer-Verlag, 1979.