Meta-Mathematics of Algebraic Complexity*

Michal Garlík
Imperial College London

Svyatoslav Gryaznov
Imperial College London

Jiaqi Lu
Imperial College London

Rahul Santhanam
University of Oxford

Iddo Tzameret
Imperial College London

Abstract

We initiate the study of the meta-mathematics of algebraic circuit lower bounds, aiming both to gain insight into the methods sufficient and necessary to prove algebraic circuit lower bounds, and to contribute to the study of bounded arithmetic as a logical foundation for complexity lower bounds. We demonstrate that while algebraic circuit lower bounds are hard for somewhat weak proof systems such as Polynomial Calculus Resolution (PCR), contemporary lower bounds are efficiently provable in proof systems and bounded arithmetic theories corresponding to NC^2, such as VNC^2 and the corresponding class of propositional Frege proofs of quasipolynomial-size. Moreover, going below VNC^2 into algebraic constant-depth reasoning is likely insufficient to efficiently prove already constant-depth algebraic circuit lower bounds. Under an algebraic analogue of the Krajíček–Razborov conjecture, our results suggest that proving superpolynomial lower bounds for general algebraic circuits will require stronger reasoning than that used in the rank-based proofs studied here. Specifically, we show the following.

NC^2-reasoning and rank method Algebraic circuit lower bounds are often proved via the "rank method", with recent prominent applications including the constant-depth lower bounds of Limaye, Srinivasan and Tavenas [LST25] and Forbes [For24]. We show that these rank-based arguments can be formalised in the bounded arithmetic theory VNC^2, which captures reasoning with NC^2 concepts. This complements the work of Tzameret and Cook [TC21], who formalised structural *upper* bounds in this theory, and provides a unified framework for studying barriers to current algebraic complexity methods, complementing barriers studied by Efremenko, Garg, Makam, Oliveira, and Wigderson [EGdOW18, GMdOW19].

Sparsity algebraic reasoning We show that Polynomial Calculus Resolution (PCR) cannot efficiently prove superpolynomial algebraic circuit lower bounds for any family of polynomials. Moreover, PCR cannot efficiently prove exponential constant-depth circuit lower bounds for any family of polynomials.

Constant-depth algebraic reasoning We introduce the Tensor Rank Principle and demonstrate it is hard for PCR. We show that if this principle is hard against constant-depth Ideal Proof System (IPS), then constant-depth IPS cannot efficiently prove constant-depth algebraic circuit lower bounds.

*A preliminary version of this work appeared in the proceedings of ACM-IEEE LICS 2026 [GGL+26]. The work of Michal Garlík, Svyatoslav Gryaznov, Jiaqi Lu, and Iddo Tzameret was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002742, *EPRICOT* project). The work of Jiaqi Lu, Rahul Santhanam, and Iddo Tzameret was also supported by the Engineering and Physical Sciences Research Council (EPSRC) under grant EP/Z534158/1.
Emails (in author order): michal.garlik@gmail.com; svyatoslav.i.gryaznov@gmail.com; jl13023@ic.ac.uk; rahul.santhanam@cs.ox.ac.uk; i.tzameret@imperial.ac.uk.

1 Introduction

The meta-mathematics of computational complexity asks questions about the nature and limits of efficient computation, with a central focus on *provability*: which proof systems or axiom systems are necessary or sufficient to establish hardness results, such as circuit-size lower bounds and complexity-class separations.

Beyond the formal provability of complexity lower bounds, one can also take a somewhat less formal approach by asking which "proof methods" can yield lower bounds, rather than in which formal proof system a given lower bound is provable. We call this direction *the barriers approach*, in contrast to the *meta-mathematical approach*.

In the *Boolean* setting, there has been extensive work on the barriers approach: developing tools for analysing which methods can yield strong Boolean circuit-size lower bounds, and which are ruled out by general obstacles. Classic examples include the *relativization* barrier of Baker, Gill and Solovay [BGS75] and the *natural proofs* barrier of Razborov and Rudich [RR97], both of which rule out broad families of standard approaches to proving superpolynomial lower bounds for NP.

These barriers have been highly instructive and influential, as they rule out large classes of lower-bound attempts. However, they can sometimes have an ad-hoc flavour, in that it is not always unambiguous whether a given lower-bound argument falls within their scope. For example, it is unclear what the relativization barrier says about low-depth circuit classes; similarly, in the natural proofs framework it can be difficult to identify the underlying "natural property", and to separate this identification from the task of proving that an explicit hard function satisfies the property.

In contrast, the meta-mathematical approach to complexity lower bounds goes through *formal logic* and specifically proof theory and independence results (namely, results showing that a theorem, and its negation, cannot be proved in a given formal theory of mathematics). One advantage of this approach is that barrier results that are formalised as logical independence results are more robust since they are closed under basic logical operations. For example, the natural-proofs barrier implies that there are no natural proofs of certain strong lower bounds. However, this does not exclude the following possibility: there is a natural proof of a statement A, and from A one can derive, by a non-natural argument, a strong lower bound B. Thus, the barrier does not rule out proving B in this way. Hence, a natural argument could be helpful in proving lower bounds despite the barrier, as long as it is only part of the proof. By contrast, logical barriers are formulated inside a fixed theory or proof system, and therefore rule out not only direct proofs of B, but also proofs that derive B through intermediate statements such as A, provided that the implication $A \to B$ is itself provable in the theory.

When thinking of formal theories of interest in computational complexity theory, the natural choice is *Bounded Arithmetic* theories, which are weak fragments of Peano Arithmetic with their axioms and definable functions usually restricted to a prescribed complexity class. Razborov [Raz95a, Raz95b] initiated a general investigation of the provability of Boolean circuit lower bounds in bounded arithmetic and propositional proof complexity, and proved strong unconditional limitations for restricted proof systems (e.g., the absence of efficient small-width DNF Resolution proofs of *any* superpolynomial Boolean circuit lower bound [Raz15]; see also [Raz98, Raz04a] for preceding PC degree and resolution size lower bounds, respectively). More recent developments in this direction appear in works by Chen, Li and Oliveira [LO23, CLO26, Oli25].

A central conjectural framework in this direction is due to Krajíček and Razborov [Kra04, Raz96, Raz16b]:[1]

[1] Razborov conjectured in [Raz15] that Frege cannot efficiently prove superpolynomial circuit lower bounds for any Boolean function. More specifically, [Raz15, Conjecture 1] with suitable parameters for the underlying combinatorial designs implies under some hardness assumptions that Frege cannot efficiently prove that $\mathsf{SAT} \notin \mathsf{P/poly}$. Further conjectures about the impossibility of *Extended* Frege to efficiently prove circuit lower bounds have been circulated in the proof complexity literature and discussions (cf. [Raz16a, Raz21, Kra11]).

Krajíček–Razborov Conjecture: Extended Frege cannot prove in polynomial size *any* superpolynomial Boolean circuit lower bound (for any Boolean function).

Similar to the developments described above, the study of barriers against proving *algebraic* circuit lower bounds was also considered in the past, although it remains less advanced than its Boolean counterpart. Two prominent frameworks for barriers have emerged. The *algebraic natural proofs* framework by Forbes, Shpilka, and Volk, and independently Grochow, Kumar, Saks, and Saraf [FSV18, GKSS17] develops algebraic analogues of natural properties. It is known that many[2] currently established algebraic lower bounds naturalise in this sense [FSV18]. However, despite interesting recent work [CKR+20, KRST22] it remains open whether there is a compelling algebraic hardness assumption ruling out algebraic natural proofs against general algebraic circuits,[3] analogous to the cryptographic evidence that natural properties against general Boolean circuits do not exist [RR97]. A different line of work studies barriers for *rank-based* approaches. Efremenko, Garg, Oliveira, and Wigderson [EGdOW18] establish unconditional limitations on matrix-rank methods for proving tensor- and Waring-rank lower bounds. Garg, Makam, Oliveira, and Wigderson [GMdOW19] extend these barriers to certain methods based on higher-order tensor rank and Waring rank. The limitations shown in this direction are *unconditional*, and apply to a broad class of techniques. On the other hand, these barriers apply only to tensor- and Waring-rank.

While barriers in algebraic circuit complexity have been investigated, its formal meta-mathematics remains unexplored, and in this work we initiate such a general study of "frontier" algebraic circuit lower bounds. The guiding question is:

What is the proof-theoretic (or propositional proof-complexity) strength needed to formalise and prove frontier lower bounds in algebraic complexity?

A natural test case is the family of rank-based lower bounds for constant-depth algebraic circuits computing the Permanent over finite fields, proved by Limaye–Srinivasan–Tavenas and extended by Forbes [LST25, For24]. These results are among the strongest unconditional lower bounds currently known for algebraic circuits, and their proofs use a robust and conceptually clean method (relative rank and set-multilinearization), making them particularly suitable for our analysis.

1.1 The rank method and constant-depth algebraic lower bounds

Following the terminology of Garg *et al.* [GMdOW19, Definition 1.8], the essential feature of the rank method (which [GMdOW19] calls "matrix-rank method" to distinguish it from objects of higher-order than matrices, e.g., tensors) is the use of a linear map $\phi\colon V \to \mathrm{Mat}_{p,q}$ and the associated subadditive measure $f \mapsto \mathrm{rank}(\phi(f))$. In the circuit lower-bound setting, such a measure yields a lower bound once it is shown to be small for every polynomial computed by small circuits in the model under consideration, and large for the target polynomial.

In particular, the arguments in Limaye, Srinivasan and Tavenas, and Forbes [LST25, For24] proceed by associating to a polynomial f a "relative rank" measure (defined with respect to an appropriate partition of variables and a set-multilinear structure), and then proving two complementary statements: (i) the Permanent has large relative rank, and (ii) every polynomial

[2] One example of an algebraic circuit lower bound technique that is not natural in the sense of algebraic natural proofs is the substitution method (see [BCS97, Chapter 6]). Another is any lower bound technique that applies to tensor rank, but not border rank, since natural proofs necessarily prove lower bounds on border complexity. We thank the anonymous reviewer for pointing this out.

[3] The non-existence of algebraic natural properties against general algebraic circuits follows from the existence of *succinct* hitting set generators against algebraic circuits, but this is not an overly standard algebraic hardness assumption in any sense, and indeed is easily seen to be equivalent to the non-existence of natural properties.

computed by a small constant-depth algebraic circuit has small relative rank. The second part requires turning a circuit into an explicit representation of the relevant coefficient vector (after suitable multilinearization), and then bounding the rank of this vector by structural properties of low-depth circuits.

From a proof-theoretic viewpoint, this raises concrete formalisation tasks: one must define the rank measure inside a weak theory, prove its basic properties, and formalise the circuit-to-coefficient-vector transformation (together with the bounds needed in the rank argument). A priori, it is not obvious how much uniform reasoning is required, because the argument mixes algebraic manipulations, combinatorial counting bounds, and circuit-structured constructions.

1.2 Our results

In this work we study the efficient provability of algebraic circuit lower bounds. We consider this question in propositional and algebraic proof systems, as well as in bounded arithmetic. The latter are first-order theories that correspond to certain computational complexity classes and serve as a uniform counterpart for propositional proof systems (see [CN10, Kra95]). We discuss the consequences of our work in Section 1.4.

Summary of results We shall demonstrate that while algebraic circuit lower bounds are hard for somewhat weak proof systems such as Polynomial Calculus Resolution (PCR), they are provable (efficiently) in proof systems and theories corresponding to NC^2, such as the bounded arithmetic theory VNC^2 and the corresponding propositional proof system NC^2-Frege. Our results suggest a complementary limitation. We show that the particular rank-based constant-depth lower-bound arguments of Limaye, Srinivasan and Tavenas, and Forbes [LST25, For24] can be formalised in the bounded arithmetic theory VNC^2.

This establishes a conditional limitation on the ability of rank-based arguments to establish general algebraic circuit lower bounds. In particular, suppose an algebraic analogue of the Krajíček–Razborov conjecture holds, asserting that VNC^2 cannot prove superpolynomial lower bounds against general algebraic circuits computing polynomials of polynomial degree. Then extensions of these rank-based lower bound arguments, whose correctness remains provable in VNC^2, cannot establish such super-polynomial algebraic circuit lower bounds.

We then show that "algebraic constant-depth" reasoning is plausibly insufficient to efficiently prove constant-depth algebraic circuit lower bounds. More precisely, we show that if a property of tensor rank is hard for algebraic proofs operating with constant-depth circuits (formally, constant-depth IPS), then so are constant-depth algebraic circuit lower bounds.

This picture—depicted in Table 1—aligns with the working assumption that rank arguments, and rank-based lower bounds specifically, sit proof-theoretically (or proof complexity-wise) in reasoning systems that operate with concepts in NC^2 (or more precisely, the complexity class DET, that is the AC^0-closure of the integer determinant; see [TC21] for a discussion and references).

1.2.1 VNC^2 and NC^2-Frege suffice

Our first contribution is to show that the entire rank-based lower bound argument against constant-depth algebraic circuits of Limaye *et al.* and Forbes [LST25, For24] can be carried out in the bounded arithmetic theory VNC^2. Using the standard translation between bounded arithmetic and propositional proofs (see, e.g., Cook–Nguyen [CN10]), we obtain the corresponding propositional upper bound.

Theorem 1.1 (Informal; Theorem 5.58 and Theorem 5.59)**.** *There are polynomial-size propositional NC^2-Frege proofs of the superpolynomial lower bound for the Permanent polynomial against constant-depth algebraic circuits over finite fields. Moreover, this lower bound is provable in the theory VNC^2.*

Theory or proof system	Algebraic circuit lower bounds
NC^2 **reasoning** VNC^2 and NC^2-Frege	**Efficiently provable.** Rank-based constant-depth lower bounds for the Permanent over finite fields are provable in VNC^2, giving polynomial-size NC^2-Frege and quasipolynomial-size Frege proofs.
Constant-depth IPS Over $\mathbb{F}_2$	**Conditionally hard.** Constant-depth algebraic circuit lower bounds require superpolynomial-size proofs, assuming exponential-size constant-depth IPS lower bounds for the tensor rank principle.
PCR Over $\mathbb{F}_2$	**Unconditionally hard.** Superpolynomial algebraic circuit lower bounds require superpolynomial-size PCR proofs.

Table 1: Three levels of reasoning and the status of algebraic circuit lower bounds.

To establish this, we provide VNC^2-formal definitions and proofs for the key components of the rank-method framework in the parameter regime used by [LST25, For24]. In particular, we formalise:

- the definition of the relevant relative-rank measure and its basic algebraic properties;
- the fact that the specific "word polynomials" used in the argument have high rank;
- a uniform procedure that, given a constant-depth algebraic circuit of bounded syntactic-degree, computes the corresponding coefficient vector needed for the rank analysis;
- the set-multilinearization step in [For24]: every constant-depth circuit can be converted into a set-multilinear circuit with only a linear increase in depth and a moderate size blow-up;
- the final rank upper bound for constant-depth set-multilinear circuits, which implies a circuit-size lower bound for the specific "word polynomials".

Conceptually, the theorem supports the informal correspondence between rank-based lower bounds and NC^2-level reasoning: the proof of [LST25, For24] does not require the strength of powerful systems such as Extended Frege.

Theorem 1.1 gives an upper bound for the proof complexity of the circuit lower bounds shown in [LST25, For24]. We believe that the upper bound is close to being tight (with respect to the proof system), which is consistent with the constant-depth algebraic analogues of the Krajíček–Razborov conjecture.

Proof overview for Theorem 1.1 We shall work with *first-order* theories of bounded arithmetic, which are translatable to polynomial-size propositional proofs. Specifically, we work with the theory VNC^2 as developed in Cook and Nguyen [CN10] and used in Tzameret and Cook [TC21] to formalise basic linear algebra in a seemingly minimal theory. Any formula of low logical complexity (formally, $\forall\Sigma^B_0$-formulas) that has a proof in VNC^2 translates naturally (by instantiating the universal variables in it by constants) to a family of propositional tautologies with polynomial-size propositional proofs in NC^2-Frege.

The formalisation of the lower bound proceeds along the lines of [LST25] (which works over fields of large characteristic), except where we use Forbes' idea for finite fields [For24]. Specifically, this means that we use a direct set-multilinearization step using the Binet–Minc [Min79] identities. We note that for simplicity of presentation, the upper bound in NC^2-Frege we demonstrate is not precisely for $\mathsf{AlgCircuit}(\mathrm{perm}_n, s, \ell, \Delta)$ as is used in Theorem 1.2, but a slightly different formulation (though it can be made to match precisely our original propositional formula).

There are three main challenges in the formalisation of the lower bounds in the theory, as follows.

Irrational numbers: We need to encode and reason about irrational numbers and functions on reals, such as exp and ln, as in [LST25], while the theory cannot do this directly. However, Jeřábek [Jeř23] has formalised these functions in the theory VTC^0, which is contained in VNC^2. Moreover, Jeřábek proved that VTC^0 can prove some properties about these functions, such as $\exp(z+w) = \exp(z)\exp(w)$ and $\exp(\log z) = z$, and these properties are sufficient for our formalisation.

Diophantine approximation: We need to use the following Diophantine approximation from [LST25]: for $a, b \in \mathbb{Z}$ with $(a,b) \neq (0,0)$, $|a\alpha - b| \geq \frac{1}{4|a\alpha|+2}$, where $\alpha = 1/\sqrt{2}$. We show that to use this approximation, it is sufficient to prove the following statement in the theory: $x^2 = 2 \ \wedge \ (a,b) \neq (0,0) \implies ax \neq b$, which in itself expresses the irrationality of x. This statement can be proved in the theory by showing that the greatest common divisor of a and b is definable in VNC^2 using the number-maximisation axiom [HJ18].

Properties of relative rank: The lower bound in [LST25] is a rank argument and hence uses several properties of matrix rank. Mulmuley [Mul86] showed how to compute the rank of a matrix by using the determinant. Following the work of Tzameret and Cook [TC21] on linear algebra in VNC^2, Ken and Kuroda [KK23] used Mulmuley's algorithm to define in the theory the rank function. Using their rank function, the required properties of the rank function can be proved in VNC^2.

1.2.2 Polynomial Calculus Resolution lower bounds

We study the hardness of the statements $\mathsf{AlgCircuit}(f, s, \ell)$ stating that a polynomial f has an algebraic circuit of size s and syntactic-degree ℓ. We also study analogous statements $\mathsf{AlgCircuit}(f, s, \ell, \Delta)$, where the circuits are restricted to depth Δ. Similar formulas were considered for Boolean circuits [Raz98, Raz04a, Raz04b, Kra04, Raz15, GGRT26, AR23] in the case of Resolution, Polynomial Calculus, Polynomial Calculus Resolution and Sum-of-Squares.

We define $\mathsf{AlgCircuit}(f, s, \ell)$ in a similar fashion to the lower bound statement from Razborov [Raz15]. We prove its hardness against PCR over the two-element field by reducing (the negation of) the *rank principle* to $\mathsf{AlgCircuit}(f, s, \ell)$. Informally, the (weak) rank principle states that an $m \times m$ matrix A cannot be represented as a product of an $m \times n$ matrix and an $n \times m$ matrix when $m > n$ and the rank of A exceeds n. Garlík, Gryaznov, Ren and Tzameret [GGRT26] studied the complexity of different encodings of the rank principle, including its *iterated* variant, which is the variant we use in the reduction. This allows us to show that PCR cannot efficiently prove lower bounds on the algebraic circuit complexity of an n-variable polynomial f when $s \geq (\ell n)^{\Omega(1)}$.

Below we use $\mathsf{PCR}_{\mathbb{F}_2}$ to denote PCR refutations over $\mathbb{F}_2$.

Theorem 1.2 (PCR cannot efficiently prove algebraic circuit lower bounds; Theorem 3.7)**.** *There exists a constant $c > 0$ such that given any polynomial f of degree at most ℓ over $\mathbb{F}_2$ in n variables and $s \geq (\ell n)^c$, any $\mathsf{PCR}_{\mathbb{F}_2}$ refutation of $\mathsf{AlgCircuit}(f, s, \ell)$ requires superpolynomially many monomials.*

This establishes a weaker version of an algebraic analogue of the Krajíček–Razborov conjecture by showing that there are no polynomial-size PCR proofs of superpolynomial algebraic circuit lower bounds for f, for *any* polynomial f.

As far as we are aware, Theorem 1.2 is the first result showing unconditional hardness of proving an algebraic circuit lower bound.

Constant-depth algebraic circuits form a strictly weaker computational model than general algebraic circuits. Consequently, establishing lower bounds against constant-depth algebraic circuits [LST25, For24] is expected to be easier than proving lower bounds against general algebraic circuits. It follows that proving proof-complexity lower bounds for formulas expressing lower

bounds against constant-depth algebraic circuits is *stronger* than proving proof-complexity lower bounds for formulas expressing lower bounds against general algebraic circuits. By reducing to the rank principle, we obtain a superpolynomial lower bound for exponential constant-depth algebraic circuit lower-bound formulas against PCR:

Theorem 1.3 (PCR cannot efficiently prove exponential constant-depth algebraic circuit lower bounds;Theorem 3.8)**.** *For every constant $c > 0$, given a polynomial f of degree at most ℓ over $\mathbb{F}_2$ in n variables and $s \geq N^{1/c}$, where N is the number of monomials in n variables of degree at most ℓ, every $\mathsf{PCR}_{\mathbb{F}_2}$ refutation of $\mathsf{AlgCircuit}(f, s, \ell, O(c))$ requires superpolynomially many monomials.*

1.2.3 Conditional constant-depth IPS lower bounds

While in the previous section we discussed the relatively weak PCR proof system, here we shall deal with a stronger system: the Ideal Proof System (IPS) introduced by Grochow and Pitassi [GP18], and specifically its constant-depth fragment. Intuitively, one should think about it roughly as an algebraic proof system similar in strength to polynomial calculus in which each proof-line is written as an algebraic constant-depth circuit, instead of as a sum of monomials (as in PCR). Alternatively, one can consider constant-depth IPS over a finite field of constant characteristic as constant-depth Frege with counting gates modulo p (augmented with the ability to freely prove [arithmetised] polynomial identities).

We consider the generalisation of the rank principle (see above), called the *tensor rank principle*, denoted $\mathsf{TRankP}^m_n(r, A)$. It states that an r-tensor $A \in \mathbb{F}^{m \times \cdots \times m}$ does not admit a tensor-rank decomposition of rank at most n, which is smaller than m. When $r = 2$, this principle coincides with the rank principle. We show a conditional superpolynomial lower bound on $\mathsf{AlgCircuit}(f, s, n\ell, \Delta)$ against constant-depth IPS. We construct a substitution that converts a short refutation of $\mathsf{AlgCircuit}(f, s, n\ell, \Delta)$ into a short refutation of an order-n tensor rank instance over $\mathbb{F}_2$, using ideas from Raz [Raz04a].

Theorem 1.4 (Constant-depth IPS cannot efficiently prove constant-depth algebraic circuit lower bounds conditionally; See Theorem 4.3 and Corollary 4.4)**.** *Consider a family of polynomials $f \in \mathbb{F}_2[u_1, \ldots, u_n]$ of degree at most ℓ, where $n \geq 1$ and $\ell \geq 2$. Set*

$$D = n\ell, \qquad N = \binom{n+\ell}{\ell}, \qquad R = \binom{n+D}{D}, \qquad K = \lfloor \sqrt{s} \rfloor.$$

Assume $K \to \infty$, $K < N$, $\Delta > 4$, and $s \geq 4n^2(\ell+1)^4$. Let A be the diagonal order-n tensor of dimension N whose $(M, \ldots, M)$th entry is $\mathrm{coeff}_M(f)$, with all other entries zero. Suppose that, for some fixed $\varepsilon > 0$ and constant $d > 5$, the corresponding systems $\mathsf{TRankP}^N_K(n, A)$ require d-IPS refutations over $\mathbb{F}_2$ of size at least 2^{K^ε}, and that

$$\log(sn) + n \log N + D \log R = o(K^\varepsilon).$$

Then $\mathsf{AlgCircuit}(f, s, D, \Delta)$ requires $(d-5)$-IPS refutations over $\mathbb{F}_2$ of size $|\mathsf{AlgCircuit}(f, s, D, \Delta)|^{\omega(1)}$.

Tensor rank lower bounds are notoriously difficult to prove, and even basic mathematical questions about tensor rank remain poorly understood. Since tensors have been studied in mathematics for much longer than algebraic circuits, there is arguably stronger empirical evidence for the difficulty of proving tensor rank lower bounds than for the difficulty of proving algebraic circuit lower bounds. Thus, grounding the formal difficulty of reasoning about algebraic circuits in the difficulty of reasoning about tensors seems to provide meaningful evidence for the hardness of proving algebraic circuit lower bounds (this was pointed out by the anonymous reviewer).

To complement our lower bounds, we show an unconditional lower bound on $\mathsf{TRankP}^m_n(r, A)$ against $\mathsf{PCR}_{\mathbb{F}_2}$. This is achieved using a random restriction argument combined with a degree-reduction to the pigeonhole principle.

Theorem 1.5 (Theorem 4.2)**.** *Let m, n, r be positive integers with $m > n$. Let $J \subseteq [m]$ with $|J| > n$ and let A be the r-tensor defined as*

$$A_{i_1,\dots,i_r} = \begin{cases} 1 & \text{if } i_1 = \cdots = i_r \in J, \\ 0 & \text{otherwise.} \end{cases}$$

Then any $\mathsf{PCR}_{\mathbb{F}_2}$ *refutation of* $\mathsf{TRankP}_n^m(r, A)$ *requires* $2^{\Omega(n)}$ *monomials.*

1.3 Relation to other work

Santhanam and Tzameret [ST0], and later Lu *et al.* [LST26], investigated the proof complexity of statements asserting algebraic circuit lower bounds. However, these works did not directly study the hardness of the algebraic circuit lower-bound statements themselves. Instead, they considered a more meta-level question: whether one can prove lower bounds for the *statement* that there are no short proofs of algebraic circuit lower bounds. This meta-level lower bound was then used as an ingredient in a diagonalisation argument.

Garlík, Gryaznov, Ren, and Tzameret [GGRT26] studied two related questions. In the first, they established a $\mathsf{PCR}_{\mathbb{F}_2}$ lower bound for noncommutative algebraic branching program (ncABP) lower bound statements. The model of ncABP is quite a weak model of computation, and further the result is specialised in the sense that the rank principle is intimately connected to the way ncABP are encoded. The second related question was the introduction of bounded arithmetic theories augmented with the rank principle as an axiom. They showed that *Boolean* complexity lower bounds such as Smolensky–Razborov [Smo87] lower bounds against constant-depth Boolean circuits with mod p gates can be formalised in theories corresponding to $\mathsf{AC}^0[p]$-reasoning augmented with the weak rank principle. These theories seem close to the theory VNC^2 we use in this work.

Müller and Pich [MP20] and others showed that Extended Frege (which corresponds to a proof system operating with polynomial-size circuits) can prove known Boolean circuit lower bounds. Our results are analogous to theirs, as we show that NC^2-Frege, i.e., proofs operating with NC^2 circuits can prove known algebraic circuit lower bounds (where the class NC^2 corresponds to polynomial-size algebraic circuits, due to known depth reductions; i.e., a polynomial-size algebraic circuit computing a polynomial of polynomial degree, can be converted to a depth-$O(\log^2 n)$ circuit of poly(n) size, for n the number of variables).

1.4 Discussion and conclusions

We conclude the introduction by explaining what our results suggest from the perspectives of algebraic and proof complexity.

First, from the perspective of algebraic complexity, our results show that known techniques, including frontier constant-depth algebraic circuit lower bounds such as those of Limaye, Srinivasan and Tavenas [LST25], can be formalised in a relatively weak theory corresponding to NC^2 reasoning, namely VNC^2. This should be contrasted with stronger bounded arithmetic theories, such as PV and S_2^1, which correspond to polynomial-time reasoning.

Thus, under an algebraic analogue of the Krajíček–Razborov conjecture, our results suggest a possible limitation of current methods. Suppose the algebraic analogue of Krajíček–Razborov conjecture holds, namely, VNC^2 reasoning cannot prove algebraic circuit lower bounds for NC^2 circuits, and hence, by depth reduction, for VP. Since we show that the present rank-based constant-depth lower bounds are provable already in VNC^2, any proof of substantially stronger algebraic circuit lower bounds would have to use ideas that are stronger, at least in their logical strength, than the concepts currently used in these arguments.

Second, from the perspective of proof complexity, our results contribute to three objectives:

- We provide new hard instances for proof systems of interest, in the tradition of proof complexity lower bounds. In our case these are instances that are themselves meta-mathematical statements (cf. Razborov and Raz [Raz98, Raz04a]). This can help to establish further new proof complexity lower bounds.

- Our instances are connected to the diagonalisation-based approach to proof complexity lower bounds. In the works of Santhanam and Tzameret [ST0] and Lu, Santhanam and Tzameret [LST26], strong proof complexity lower bounds were obtained for CNF formulas expressing the hardness of proving algebraic circuit lower bounds. A caveat in this approach is that it is not known, unconditionally, whether these CNF formulas are unsatisfiable. The present work identifies a condition under which such formulas are unsatisfiable: namely, the hardness of proving algebraic circuit lower bounds in constant-depth IPS. Combined with [LST26], this condition would imply lower bounds for $\mathsf{AC}^0[p]$-Frege, a longstanding open problem in proof complexity.

- Studying the unprovability of lower bounds is a major effort in proof complexity. It helps relate the strength of efficient computation to that of efficient provability, which is motivated by the goal to understand different proof systems and bounded arithmetic theories.

2 Preliminaries

To fix notation, we define Boolean circuits with fan-in two gates $\wedge, \vee$ and fan-in one gates $\neg$: for every $n \in \mathbb{N}$, an n-input, single-output Boolean circuit is a directed acyclic graph with n sources (vertices with no incoming edges) and one sink (a vertex with no outgoing edges). All non-source vertices are called gates or nodes and are labelled with one of $\wedge$, $\vee$ or $\neg$ (i.e., the logical operations OR, AND, and NOT). The vertices labelled with $\wedge$ and $\vee$ have fan-in (i.e., number of incoming edges) equal to 2, and the vertices labelled with $\neg$ have fan-in 1. The size of C, denoted by $|C|$, is the number of vertices in it. If C is a Boolean circuit and $x \in \{0,1\}^n$ is some input, then the output of C on x is denoted by $C(x)$.

We use $[n]$ to denote the set $\{1, \ldots, n\}$.

2.1 Algebraic complexity

We give a brief overview of standard notions in algebraic complexity. For a more detailed treatment, we refer the reader to [SY10].

Let $\mathbb{F}$ be a field. Denote by $\mathbb{F}[\overline{x}]$ the ring of polynomials with coefficients from $\mathbb{F}$ and variables $\overline{x} = (x_1, \ldots, x_n)$. A polynomial is a formal linear combination of monomials, where a monomial is a product of variables. Two polynomials are identical if all their monomials have the same coefficients. The degree of a polynomial is the maximum total degree of a monomial in it.

Definition 2.1 (Algebraic circuits and algebraic formulas)**.** An *algebraic circuit* over a field $\mathbb{F}$ is a finite directed acyclic graph. The leaves (nodes with in-degree zero) are called *input nodes*. Each input node is labelled either with a variable or a field element in $\mathbb{F}$. All the other nodes have in-degree two and are labelled by either $+$ (addition) or $\times$ (multiplication). The output of a $+$-node is the sum of the polynomials computed by its incoming nodes, and the output of a $\times$-node is their product. An algebraic circuit is called an *algebraic formula* if the underlying directed acyclic graph is a tree (every node has at most one outgoing edge). The *size* of an algebraic circuit C is the number of nodes in it, denoted by $|C|$. The *depth* of C is the length of the longest directed path in it, denoted by $\mathrm{depth}(C)$. The *product-depth* of C is the maximum number of $\times$-nodes on a directed path in C. The polynomial computed by a node u in an algebraic circuit is denoted by $\widehat{C}_u$. The polynomial computed by an algebraic circuit C is denoted by $\widehat{C}$.

Definition 2.2 (Depth-Δ algebraic circuits and algebraic formulas)**.** A *depth-Δ algebraic circuit (or formula)* over a field $\mathbb{F}$ is an algebraic circuit (or formula) with unbounded fan-in of depth Δ. Except for the input nodes, all other nodes may have unbounded fan-in and are labelled by $+$ or $\times$. All the rest of the notions are defined similarly to Definition 2.1.

Definition 2.3 (Syntactic-degree $\mathrm{sdeg}(\cdot)$)**.** Let C be an algebraic circuit (with possibly unbounded fan-in) and v a node in C. The *syntactic-degree* $\mathrm{sdeg}(v)$ of v is defined as follows:

1. If v is a field element or a variable, then $\mathrm{sdeg}(v) \coloneqq 0$ and $\mathrm{sdeg}(v) \coloneqq 1$, respectively;
2. If $v = \sum_{i=0}^{t} u_i$ then $\mathrm{sdeg}(v) \coloneqq \max\{\mathrm{sdeg}(u_0), \ldots, \mathrm{sdeg}(u_t)\}$;
3. If $v = \prod_{i=0}^{t} u_i$ then $\mathrm{sdeg}(v) \coloneqq \sum_{i=0}^{t} \mathrm{sdeg}(u_i)$.

Negation is not a primitive arithmetic gate: $-g$ is computed as $(-1)g$. Since constants have syntactic degree zero, multiplication by -1 does not increase syntactic degree.

Definition 2.4 (Algebraic Branching Program)**.** An *algebraic branching program* (ABP) over a field $\mathbb{F}$ and variables $\mathbf{x} = (x_1, \ldots, x_n)$ is a directed acyclic graph $G = (V, E)$ with the following properties:

- The vertices are partitioned into layers
$$V = L_0 \cup L_1 \cup \cdots \cup L_d,$$
where $L_0 = \{s\}$ contains a unique source and $L_d = \{t\}$ contains a unique sink.
- Edges go only between consecutive layers: $E \subseteq \bigcup_{i=0}^{d-1} L_i \times L_{i+1}$.
- Each edge $e \in E$ is labelled by an affine linear form $\ell_e(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_n]$.

The ABP computes the polynomial
$$f(\mathbf{x}) \;=\; \sum_{P: s \to t} \prod_{e \in P} \ell_e(\mathbf{x}),$$
where the sum is taken over all directed paths from s to t. The *size* of the ABP is $|V| + |E|$, and the *width* is $\max_i |L_i|$.

Definition 2.5 (VP [Val79])**.** Over a field $\mathbb{F}$, $\mathsf{VP}_{\mathbb{F}}$ is the class of families $f = (f_n)_{n=1}^{\infty}$ of polynomials f_n such that f_n has $\mathrm{poly}(n)$ input variables, has degree $\mathrm{poly}(n)$, and can be computed by algebraic circuits over $\mathbb{F}$ of $\mathrm{poly}(n)$ size.

Definition 2.6 (VNP [Val79])**.** Over a field $\mathbb{F}$, $\mathsf{VNP}_{\mathbb{F}}$ is the class of families $g = (g_n)_{n=1}^{\infty}$ of polynomials g_n such that g_n has $\mathrm{poly}(n)$ input variables and has degree $\mathrm{poly}(n)$, and can be written as
$$g_n(x_1, \ldots, x_{\mathrm{poly}(n)}) = \sum_{\overline{e} \in \{0,1\}^{\mathrm{poly}(n)}} f_n(\overline{e}, \overline{x})$$
for some family $(f_n) \in \mathsf{VP}_{\mathbb{F}}$.

Definition 2.7 (VAC^0)**.** Over a field $\mathbb{F}$, $\mathsf{VAC}^0_{\mathbb{F}}$ is the class of families $f = (f_n)_{n=1}^{\infty}$ of polynomials f_n such that f_n has $\mathrm{poly}(n)$ input variables, has degree $\mathrm{poly}(n)$, and can be computed by algebraic circuits over $\mathbb{F}$ of $\mathrm{poly}(n)$ size and depth $O(1)$.

Notice that VP, VNP and VAC^0 are all non-uniform complexity classes.

Definition 2.8 (Projection reduction [Val79])**.** A polynomial $f(x_1, \ldots, x_n)$ is a projection of a polynomial $g(y_1, \ldots, y_m)$ if there is a mapping σ from $\{y_1, \ldots, y_m\}$ to $\{0, 1, x_1, \ldots, x_n\}$ such that $f(x_1, \ldots, x_n) = g(\sigma(y_1), \ldots, \sigma(y_m))$. A family of polynomials (f_n) is a polynomial projection or p-projection of another family (g_n) if there is a function $t(n) = n^{\Theta(1)}$ such that f_n is a projection of $g_{t(n)}$ for all (sufficiently large) n. We say that f is projection-reducible to g if f is a projection of g.

The symmetric group over n elements $\{1, \ldots, n\}$, denoted by S_n, is the group consisting of all the bijective functions from $[n]$ to $[n]$ whose group operation is that of function composition. The sign $\operatorname{sgn}(\sigma)$ of a permutation $\sigma \in S_n$ is 1 if σ can be expressed as a product of an even number of transpositions (exchanges of two, not necessarily consecutive, entries), and -1 otherwise.

Definition 2.9 (Determinant)**.** The Determinant of an $n \times n$ matrix $A = (a_{i,j})$ is defined as

$$\det(A) \coloneqq \sum_{\sigma \in S_n} (\operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}).$$

Definition 2.10 (Permanent)**.** The Permanent of an $n \times n$ matrix $A = (a_{i,j})$ is defined as

$$\operatorname{perm}(A) \coloneqq \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i,\sigma(i)}.$$

Theorem 2.11 ([Val79])**.** *For every field $\mathbb{F}$, every polynomial family on n variables that is computable by an algebraic formula of size u is projection-reducible to the Determinant polynomial (over the same field) on $u + 2$ variables. For every field $\mathbb{F}$ of characteristic different from 2, every polynomial family in $\mathsf{VNP}_{\mathbb{F}}$ is projection-reducible to the Permanent polynomial (over the same field) with polynomially more variables.*

It is known that the Permanent polynomial is complete under p-projections for VNP when the field $\mathbb{F}$ has characteristic different from 2. The Determinant polynomial is not known to be complete for VP under p-projections.

Definition 2.12 (Iterated Matrix Multiplication)**.** Let n and d be positive integers. The *Iterated Matrix Multiplication* $\mathrm{IMM}_{n,d}$ on $N = dn^2$ variables is defined as follows. The underlying variables are partitioned into d sets $X_1, \ldots, X_d$ of size n^2, each of which is represented as an $n \times n$ matrix with distinct variable entries. Then $\mathrm{IMM}_{n,d}$ is defined to be the $(1, 1)$th entry of the matrix product $X_1 \cdot X_2 \cdots X_d$.

The Iterated Matrix Multiplication is in VP.

Theorem 2.13 (Superpolynomial lower bounds against constant-depth circuits over a field of large characteristic [LST25])**.** *Assume $d \leq \frac{\log n}{100}$ and the characteristic of $\mathbb{F}$ is 0 or greater than d. For any product-depth $\Delta \geq 1$, any algebraic circuit C computing $\mathrm{IMM}_{n,d}$ of product-depth at most Δ must have size at least n^{d^ε} where $\varepsilon = \exp(-O(\Delta))$.*

A recent result by Forbes [For24] extended this result to *any* field, including finite fields. The lower bounds in [LST25] were improved in [BDS24] and were later extended by Forbes [For24] to arbitrary fields. However, in this paper, we will focus on formalising proofs in [LST25] and [For24].

Corollary 2.14 (Superpolynomial lower bounds on constant-depth circuits over any field [For24])**.** *The lower bound of Theorem 2.13 holds over any field.*

Since $\mathrm{IMM}_{n,d}$ is a p-projection of the Permanent polynomial with $\operatorname{poly}(n, d)$ many variables, it follows that the Permanent does not have constant-depth polynomial-size circuits over any field, in the following sense. Recall that a *symbolic $n \times n$ matrix* is the $n \times n$ matrix with n^2 distinct indeterminates x_{ij}, for $i, j \in [n]$.

Theorem 2.15 (No polynomial-size constant-depth circuits for the Permanent)**.** *Let $\Delta \geq 1$ and let $\mathbb{F}$ be a field. There are no constants c_1, c_2 such that the Permanent polynomial $\operatorname{perm}(A)$ of the $n \times n$ symbolic matrix A over the field $\mathbb{F}$ is computable by an algebraic circuit of size $c_1 n^{c_2}$ and depth Δ, where Δ is independent of n, for all sufficiently large n.*

2.1.1 Universal algebraic circuits

For general algebraic circuits, we have the following.

Theorem 2.16 (Existence of universal circuits for homogeneous polynomials [Raz10])**.** *Let $\mathbb{F}$ be a field and let $\overline{x}$ consist of n variables, and let $C^{\mathrm{hom}}_{t,d}$ denote the class of all homogeneous polynomials of total degree exactly d in $\mathbb{F}[\overline{x}]$ that have algebraic circuits of size at most t. Then there is a circuit $U(\overline{x},\overline{w}) \in \mathbb{F}[\overline{x},\overline{w}]$ of size $O(d^2t^8)$ and syntactic-degree $O(d)$ with $K_{t,d} = O(d^2t^8)$ auxiliary variables $\overline{w}$ disjoint from $\overline{x}$, that is universal for $C^{\mathrm{hom}}_{t,d}$ in the following sense: if $f(\overline{x}) \in C^{\mathrm{hom}}_{t,d}$, then there exists $\overline{a} \in \mathbb{F}^{K_{t,d}}$ such that $U(\overline{x},\overline{a}) = f(\overline{x})$. Moreover, given t and d as input, the number $K_{t,d}$ can be computed in time $\mathrm{poly}(t,d)$. We use $K_{t,d}$ to denote this number of auxiliary variables.*

We use the following simple adaptation from [ST21]:

Definition 2.17 (Universal circuits for polynomials [ST21])**.** The universal circuit for polynomials of degree at most d and circuit size at most t is defined as:

$$U(\overline{x},\overline{w}) = \sum_{i=0}^{d} U_i(\overline{x},\overline{w}),$$

where $U_i(\overline{x},\overline{w})$ is the universal circuit for the class $C^{\mathrm{hom}}_{t,i}$ of homogeneous degree-i polynomials in $\mathbb{F}[\overline{x}]$ and where the $\overline{w}$-variables in each distinct $U_i(\overline{x},\overline{w})$ are pairwise disjoint.

The size of $U(\overline{x},\overline{w})$ is $\sum_{i=0}^{d} O(i^2t^8) = O(d^3t^8)$.

Theorem 2.18 (Existence of constant-depth universal circuits for polynomials computed by constant-depth circuits [LST26])**.** *Let $\mathbb{F}$ be a field and let $\overline{x}$ consist of n variables, and let $\mathsf{VAC}^0_{t,d}$ denote the class of all polynomials in $\mathbb{F}[\overline{x}]$ that have algebraic circuits of size at most t and depth at most d where t is polynomially bounded and d is constant. Then there is a circuit $U(\overline{x},\overline{w}) \in \mathbb{F}[\overline{x},\overline{w}]$ of size $\mathrm{poly}(t)$ and constant-depth with K auxiliary variables $\overline{w}$ disjoint from $\overline{x}$, that is universal for $\mathsf{VAC}^0_{t,d}$ in the following sense: if $f(\overline{x}) \in \mathsf{VAC}^0_{t,d}$, then there exists $\overline{a} \in \mathbb{F}^K$ such that $U(\overline{x},\overline{a}) = f(\overline{x})$. Observe that given t and d, K can be computed efficiently and is bounded by $\mathrm{poly}(t)$. We use $K_{t,d}$ to denote the number K computed from t and d.*

2.1.2 Extracting coefficients and IPS refutation formula

Let $f(\overline{x},\overline{w}) \in \mathbb{F}[\overline{x},\overline{w}]$ be a polynomial, and let $M = \prod_{i\in I} x_i^{\alpha_i} \cdot \prod_{j\in J} w_j^{\beta_j}$ be a monomial in $f(\overline{x},\overline{w})$, for some $\alpha_i, \beta_i \in \mathbb{N}$. We call $\sum_{i\in I} \alpha_i$ the $\overline{x}$-degree of M.

Definition 2.19 ($\mathrm{coeff}_M(\cdot)$ [ST21])**.** Let $f(\overline{x},\overline{w})$ be a polynomial over $\mathbb{F}$ in the disjoint sets of variables $\overline{x},\overline{w}$. Let M be an $\overline{x}$-monomial of degree j. Viewing f as a polynomial in $\mathbb{F}[\overline{w}][\overline{x}]$, $\mathrm{coeff}_M(f(\overline{x},\overline{w}))$ is defined as the (polynomial) coefficient from $\mathbb{F}[\overline{w}]$ (that is, in the $\overline{w}$-variables only) of M in $f(\overline{x},\overline{w})$.

Observe that $f(\overline{x},\overline{w}) = \sum_{M_i} M_i \cdot \mathrm{coeff}_{M_i}(f(\overline{x},\overline{w}))$, where the M_i are all possible $\overline{x}$-monomials of degree at most d, for d the maximum $\overline{x}$-degree of a monomial in $f(\overline{x},\overline{w})$.

Proposition 2.20 (Computation of coefficients in general circuits [ST21])**.** *Let $f(\overline{x},\overline{w})$ be a polynomial over $\mathbb{F}$ in the disjoint sets of variables $\overline{x},\overline{w}$. Suppose that M is an $\overline{x}$-monomial of degree d, and assume that there is an algebraic circuit computing $f(\overline{x},\overline{w})$ of size s and syntactic-degree ℓ. Then, there is a circuit of size $O(7^d \cdot s)$ computing $\mathrm{coeff}_M(f(\overline{x},\overline{w}))$ of syntactic-degree $\ell^{O(1)}$.*

Proposition 2.21 (Computation of coefficients in constant-depth circuits)**.** *Let $f(\overline{x},\overline{w})$ be a polynomial over $\mathbb{F}$ in the disjoint sets of variables $\overline{x},\overline{w}$. Suppose that M is an $\overline{x}$-monomial of degree d, and assume that there is an algebraic circuit $C(\overline{x},\overline{w})$ computing $f(\overline{x},\overline{w})$ of maximum multiplication fan-in t, size s, syntactic-degree ℓ and depth Δ, where Δ is a constant such that*

1. *all edges from the leaves are to + nodes;*

2. *all output-nodes are + nodes;*

3. *the nodes of C are alternating: that is, if v is a + node and (u, v) is an edge, then u is a $\times$ node, and if v is a $\times$ node and (u, v) is an edge, then u is either a leaf or a + node.*

Then there is a constant-depth algebraic circuit of depth Δ and size $O(2^{(t+d)d} \cdot s)$ computing $\mathrm{coeff}_M(f(\overline{x}, \overline{w}))$ of syntactic-degree $\ell^{O(1)}$.

2.1.3 Encoding of algebraic circuit upper bound formulas

Definition 2.22 (Algebraic circuit upper bound formula)**.** Let $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ be a polynomial in n variables of degree at most ℓ. The following set of polynomial equations $\mathsf{AUB}(f, s, \ell)$ (in the $\overline{w}$-variables only) expresses that the polynomial $f(\overline{x})$ can be computed by an algebraic circuit of size s:

$$\left\{\mathrm{coeff}_{M_i}(\widehat{U}(\overline{x}, \overline{w})) = b_i : 1 \leq i \leq N\right\},$$

where $\overline{b} = \mathrm{coeff}(f(\overline{x})) \in \mathbb{F}^N$ is the N-dimensional coefficient vector of the polynomial f, $U(\overline{x}, \overline{w})$ is the universal circuit for polynomials of degree at most ℓ and circuit size at most s, $\overline{w}$ are the auxiliary edge variables of U, $\{M_i\}_{i=1}^N$ is the set of all possible $\overline{x}$-monomials of degree at most ℓ, and $N = \sum_{j=0}^{\ell} \binom{n+j-1}{j} = 2^{O(n+\ell)}$ is the number of monomials of total degree at most ℓ over n variables. The size of this set of polynomial equations is $O(7^\ell \ell^3 s^8 N)$.

Definition 2.23 (Bounded-depth algebraic circuit upper bound formula)**.** Let $f(\overline{x})$ be a polynomial over $\mathbb{F}$ in n variables $\overline{x}$ of degree at most ℓ. The following set of polynomial equations $\mathsf{AUB}(f, s, \ell, \Delta)$ (in the $\overline{w}$-variables only) expresses that the polynomial $f(\overline{x})$ can be computed by an algebraic circuit of size s and depth Δ:

$$\left\{\mathrm{coeff}_{M_i}(\widehat{U}(\overline{x}, \overline{w})) = b_i : 1 \leq i \leq N\right\},$$

where $\overline{b} = \mathrm{coeff}(f(\overline{x})) \in \mathbb{F}^N$ is the N-dimensional coefficient vector of the polynomial f, $U(\overline{x}, \overline{w})$ is the bounded-depth universal circuit for polynomials computable by circuits of depth at most Δ and size at most s, $\overline{w}$ are the $K_{s,\Delta}$ edge variables, $\{M_i\}_{i=1}^N$ is the set of all possible $\overline{x}$-monomials of degree at most ℓ, and $N = \sum_{j=0}^{\ell} \binom{n+j-1}{j} = 2^{O(n+\ell)}$ is the number of monomials of total degree at most ℓ over n variables. The size of this set of polynomial equations is $O(2^{(t+\ell)\ell} \cdot \mathrm{poly}(s) \cdot N)$, where t is the maximum multiplication fan-in in $U(\overline{x}, \overline{w})$.

Definition 2.24 (Straight Line Program (SLP))**.** An SLP of a circuit $C(\overline{x})$, denoted by $\mathsf{SLP}(C(\overline{x}))$, is a sequence of equations between variables such that, assuming all the equations hold, the extension variable corresponding to the output node equals the value computed by the circuit. Formally, we choose any topological order $g_1, g_2, \ldots, g_i, \ldots, g_{|C|}$ on the nodes of the circuit C (that is, if there is a directed path from g_j to g_k in C, then $j < k$) and define the following set of equations to be the SLP of $C(\overline{x})$:

$g_i = g_{j1} \circ g_{j2} \circ \cdots \circ g_{jt}$ for $\circ \in \{+, \times\}$ iff g_i is a $\circ$ node in C with t incoming edges from $g_{j1}, \ldots, g_{jt}$.

An SLP representation of a circuit equation $C(\overline{x}) = 0$ means that we add to the SLP above the equation $g_{|C|} = 0$, where $g_{|C|}$ is the output node of the circuit.

2.2 Proof complexity

Definition 2.25 (Polynomial Calculus Resolution [ABSRW02])**.** Let $\mathbb{F}$ be a field. *Polynomial Calculus Resolution over $\mathbb{F}$ ($\mathsf{PCR}_\mathbb{F}$)* is a proof system whose lines are polynomials in $\mathbb{F}[x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}]$, where $\overline{x_1}, \ldots, \overline{x_n}$ are treated as new formal variables. PCR has

Boolean axioms for all variables, including the variables $\overline{x_i}$, together with the additional axioms $x_i + \overline{x_i} = 1 : i \in [n]$. Each line is either an axiom or is derived from the previous lines using the following inference rules:

$$\frac{f \qquad g}{\alpha f + \beta g} \quad \text{and} \quad \frac{f}{x \cdot f},$$

where $\alpha, \beta \in \mathbb{F}$ are any scalars and x is a variable.

For a clause C, denote by Γ_C the monomial

$$\Gamma_C \coloneqq \prod_{\overline{x} \in C} x \cdot \prod_{x \in C} \overline{x}$$

and for a CNF τ, let $\Gamma_\tau \coloneqq \{\Gamma_C : C \in \tau\}$. Note that τ is unsatisfiable if and only if the polynomials in Γ_τ, together with the Boolean and complementarity axioms of PCR, have no common root. A PCR *refutation* of a CNF τ is a PCR proof of the contradiction $1 = 0$ from Γ_τ.

The *degree* of a PCR proof is defined as the maximum degree of a polynomial appearing in it, and its *size* is the number of distinct *monomials* appearing in the proof.

It is convenient to reduce all polynomials appearing in PCR proofs modulo the ideal generated by the Boolean axioms. This makes the proof lines multilinear, and can only decrease their size and degree.

We shall also use the system NC^2-Frege, which is an inference-based propositional proof system whose proof lines are represented by NC^2-circuits. These are fan-in-two De Morgan circuits of depth $O(\log^2 n)$, where n is the number of input variables. We refer the reader to [HT15] for a formal definition of this system.

Ideal Proof Systems Given $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ over some field $\mathbb{F}$, Hilbert's Nullstellensatz shows that the polynomial system $f_1(\overline{x}) = \cdots = f_m(\overline{x}) = 0$ is unsatisfiable (over the algebraic closure of $\mathbb{F}$) if and only if there are polynomials $g_1, \ldots, g_m \in \mathbb{F}[\overline{x}]$ such that $\sum_{j=1}^m g_j(\overline{x}) f_j(\overline{x}) = 1$ (as a formal identity), or equivalently, that 1 is in the ideal generated by $\{f_j\}_{j=1}^m$ (we use the notation from [FSTW21]).

Definition 2.26 ((Boolean) Ideal Proof System (IPS) [GP18])**.** Let $f_1(\overline{x}), \ldots, f_m(\overline{x}), p(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ be a collection of polynomials. An IPS proof of $p(\overline{x}) = 0$ from $\{f_j(\overline{x})\}_{j=1}^m$, showing that $p(\overline{x}) = 0$ is semantically implied from the assumptions $\{f_j(\overline{x}) = 0\}_{j=1}^m$ over 0-1 assignments, is an algebraic circuit

$$C(\overline{x}, \overline{y}, \overline{z}) \in \mathbb{F}[\overline{x}, y_1, \ldots, y_m, z_1, \ldots, z_n],$$

such that the following equalities are formal polynomial identities.[4]

1. $C(\overline{x}, \overline{0}, \overline{0}) = 0$.
2. $C(\overline{x}, f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n) = p(\overline{x})$.

The size of the IPS proof is the size of the circuit C. The variables $\overline{y}, \overline{z}$ are sometimes called the *placeholder variables* since they are used as placeholders for the axioms. An IPS proof $C(\overline{x}, \overline{y}, \overline{z})$ of $1 = 0$ from $\{f_j(\overline{x}) = 0\}_{j=1}^m$ is called an IPS refutation of $\{f_j(\overline{x}) = 0\}_{j=1}^m$. If C comes from a restricted class of algebraic circuits $\mathcal{C}$, then such a refutation is called a $\mathcal{C}$-IPS refutation. When $\mathcal{C}$ is the class of all algebraic circuits of depth at most Δ, we write Δ-IPS to denote $\mathcal{C}$-IPS.

[4] That is, $C(\overline{x}, \overline{0}, \overline{0})$ computes the zero polynomial and $C(\overline{x}, f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n)$ computes the polynomial $p(\overline{x})$

Variable	Meaning
$\alpha_{M_i,v} : i \in [N]$	the coefficient of M_i in the polynomial computed by the gate v.
$\alpha_{M_i,v,b} : i \in [N], b \in [s]$	the coefficient of M_i in the polynomial computed by the bth input of v.
$\mathsf{Type}(v)$	0 if v is a $+$ gate and 1 if v is a $\times$ gate.
$\mathsf{InputType}_b(v) : b \in [s]$	0 if the bth input of v is a variable or a constant and 1 if it is a gate.
$\mathsf{InputType}'_b(v) : b \in [s]$	when $\mathsf{InputType}_b(v) = 0$, it is 0 if the bth input of v is a constant and 1 if it is a variable.
$\mathsf{InputType}''_b(v) : b \in [s]$	when $\mathsf{InputType}_b(v) = \mathsf{InputType}'_b(v) = 0$, it is the constant value of the bth input of v.
$\mathsf{InputVar}_b(v,j) : b \in [s], j \in [n]$	when $\mathsf{InputType}_b(v) = 0$ and $\mathsf{InputType}'_b(v) = 1$, it is 1 iff the bth input of v is the x_j.
$\mathsf{InputNode}_b(v,v') : b \in [s], v' < v$	when $\mathsf{InputType}_b(v) = 1$, it is 1 iff the bth input of v is the gate v'.

Table 2: Variables of $\mathsf{AlgCircuit}(f,s,\ell)$, for every gate $v \in [s]$.

3 Hardness of Algebraic Circuit Lower Bounds from Hardness of Weak Rank Principle

We define the *polynomial system* $\mathsf{AlgCircuit}(f,s,\ell)$ as a formalisation stating that a polynomial f can be computed by an algebraic circuit of size s and syntactic degree ℓ. Informally, this involves encoding the type of each gate (whether it is a variable, constant, $+$ gate, or $\times$ gate), the inputs of the gate, and the polynomial it computes by specifying the coefficient vector over all possible monomials of degree at most ℓ. Additionally, it is convenient to introduce auxiliary variables to represent the polynomials computed by the inputs of $+$ and $\times$ gates. The axioms of $\mathsf{AlgCircuit}(f,s,\ell)$ ensure the correctness of the encoded circuit. This definition closely resembles the lower bound formula from [Raz15].

Definition 3.1 ($\mathsf{AlgCircuit}(f,s,\ell)$)**.** Let $f(\overline{u})$ be a polynomial in n variables. Let N be the number of $\overline{u}$-monomials of degree at most ℓ and $\{M_i\}_{i\in[N]}$ be all such monomials. The variables of $\mathsf{AlgCircuit}(f,s,\ell)$ together with their intended meaning are listed in Table 2.

For every $v \in [s]$, the axioms of $\mathsf{AlgCircuit}(f,s,\ell)$ are the following polynomials:

$$(1 - \mathsf{InputType}_b(v))(1 - \mathsf{InputType}'_b(v))(\alpha_{1,v,b} - \mathsf{InputType}''_b(v)),$$

for all $b \in [s]$, making $\mathsf{InputType}''_b(v)$ consistent with $\alpha_{1,v,b}$ when the bth input of v is a constant,

$$(1 - \mathsf{InputType}_b(v))(1 - \mathsf{InputType}'_b(v))\alpha_{M_i,v,b},$$

for all $b \in [s], i \in [N], M_i \neq 1$, setting $\alpha_{M_i,v,b}$ to 0 when the bth input of v is a constant,

$$(1 - \mathsf{InputType}_b(v))\mathsf{InputType}'_b(v)\mathsf{InputVar}_b(v,j)\mathsf{InputVar}_b(v,j'),$$

for all $b \in [s], j \neq j' \in [n]$, ensuring that at most one $\mathsf{InputVar}_b(v,j)$ is set to 1 when the bth input of v is a variable,

$$(1 - \mathsf{InputType}_b(v))\mathsf{InputType}'_b(v)\left(\sum_{j\in[n]} \mathsf{InputVar}_b(v,j) - 1\right),$$

for all $b \in [s]$, ensuring that at least one $\mathsf{InputVar}_b(v,j)$ is set to 1 when the bth input of v is a variable,

$$(1 - \mathsf{InputType}_b(v))\mathsf{InputType}'_b(v)\mathsf{InputVar}_b(v,j)(\alpha_{x_j,v,b} - 1),$$

for all $b \in [s], j \in [n]$, making $\mathsf{InputVar}_b(v,j)$ consistent with $\alpha_{x_j,v,b}$ when the bth input of v is a variable,

$$(1 - \mathsf{InputType}_b(v))\mathsf{InputType}'_b(v)\mathsf{InputVar}_b(v,j)\alpha_{M_i,v,b},$$

for all $b \in [s], j \in [n], i \in [N], M_i \neq x_j$, setting $\alpha_{M_i,v,b}$ to 0 when the bth input of v is a variable,

$$\mathsf{InputType}_b(v)\mathsf{InputNode}_b(v,v')\mathsf{InputNode}_b(v,v''),$$

for all $b \in [s], v' \neq v'' < v$, ensuring that at most one $\mathsf{InputNode}_b(v,v')$ is set to 1 when the bth input of v is a gate,

$$\mathsf{InputType}_b(v)\left(\sum_{v'<v}\mathsf{InputNode}_b(v,v') - 1\right),$$

for all $b \in [s]$, ensuring that at least one $\mathsf{InputNode}_b(v,v')$ is set to 1 when the bth input of v is a gate,

$$\mathsf{InputType}_b(v)\mathsf{InputNode}_b(v,v')(\alpha_{M_i,v,b} - \alpha_{M_i,v'}),$$

for all $b \in [s], v' < v, i \in [N]$, making $\alpha_{M_i,v,b}$ and $\alpha_{M_i,v'}$ consistent when the bth input of v is a gate,

$$(1 - \mathsf{Type}(v))\left(\alpha_{M_i,v} - \sum_{b\in[s]}\alpha_{M_i,v,b}\right),$$

for all $i \in [N]$, making $\alpha_{M_i,v}$ equal to the sum of its children when v is a $+$ node,

$$\mathsf{Type}(v)\left(\alpha_{M_i,v} - \left(\sum_{\substack{i_1,\ldots,i_\ell\in[N]\\ \prod_{b\in[\ell]}M_{i_b}=M_i}}\prod_{b\in[\ell]}\alpha_{M_{i_b},v,b}\right)\prod_{b\in[s]\setminus[\ell]}\alpha_{1,v,b}\right),$$

for all $i \in [N]$, making $\alpha_{M_i,v}$ equal to the product of its children when v is a $\times$ node,

$$\mathsf{Type}(v)\prod_{b\in[\ell]}\alpha_{M_{i_b},v,b},$$

for all $i_1, \ldots, i_\ell \in [N]$ with $\deg(\prod_{b\in[\ell]} M_{i_b}) > \ell$, setting all high-degree monomials to 0,

$$\mathsf{Type}(v)\alpha_{M_i,v,b},$$

for all $b \in [s] \setminus [\ell], i \in [N], M_i \neq 1$, ensuring that only the first ℓ inputs of v are non-constant polynomials,

$$\alpha_{M_i,s} - \mathrm{coeff}_{M_i}(f),$$

for all $i \in [N]$, making $\alpha_{M_i,s}$ (the coefficient vector of the output gate) consistent with the coefficient vector of f.

The size of $\mathsf{AlgCircuit}(f,s,\ell)$ is $\mathrm{poly}(s,n,N^\ell)$.

Remark 3.2. We limit the syntactic degree of the circuit to ℓ, thus every $\times$ node can have at most ℓ inputs computing non-constant polynomials. This allows us to assume that these input gates are among the first ℓ inputs, thus making the size of the encoding significantly smaller: it suffices to consider only N^ℓ tuples of monomials instead of $\binom{N}{s}$.

We demonstrate the hardness of $\mathsf{AlgCircuit}(f,s,\ell)$ by reducing an iterated encoding of the negation of the *weak rank principle* [GGRT26] to it. The weak rank principle states that an $m \times m$ matrix of rank greater than n cannot be expressed as the product of an $m \times n$ matrix and an $n \times m$ matrix. Under our substitution, every axiom of $\mathsf{AlgCircuit}(f,s,\ell)$ has a short PCR derivation from the iterated-rank axioms; hence a short refutation of $\mathsf{AlgCircuit}(f,s,\ell)$ would yield

a short refutation of the iterated rank system. We first define the non-iterated polynomial system $\mathsf{RankP}_n^m(A)$ encoding the negation of the rank principle.

Definition 3.3 (The (Weak) Rank Principle [GGRT26])**.** Let $\mathbb{F}$ be a field and m and n two positive integers with $m > n$. We denote by $\mathsf{RankP}_n^m(A)$ the system of degree-2 polynomial equations stating that the rank of an $m \times m$ matrix A is at most n over $\mathbb{F}$. More precisely, $\mathsf{RankP}_n^m(A)$ is defined over $2mn$ variables arranged into two matrices $X \in \mathbb{F}^{m\times n}$ and $Y \in \mathbb{F}^{n\times m}$. For every $i, j \in [m]$, there is an equation in $\mathsf{RankP}_n^m(A)$ stating that the (i, j)th entry of the product XY is equal to $A_{i,j}$. That is, $\mathsf{RankP}_n^m(A)$ consists of the polynomials

$$\sum_{k\in[n]} x_{i,k}y_{k,j} - A_{i,j}, \qquad\qquad i, j \in [m].$$

By linear algebra, when the rank of A exceeds n, the system $\mathsf{RankP}_n^m(A)$ is unsatisfiable.

To perform the above reduction, we define an iterated version of $\mathsf{RankP}_n^m(A)$. Let L and n be positive integers and $\Sigma = \{0, \dots, L\}$ an alphabet of size $L+1$. We denote by $\Sigma^{\le n}$ the set of strings over Σ of length at most n. Similarly, $\Sigma^{<n}$ denotes the set of all strings over Σ of length strictly less than n. Let K be a positive integer. For every string $\pi \in \Sigma^{\le n}$, let $X^{(\pi)} = (x_{i,k}^{(\pi)})_{i\in[(L+1)K],k\in[K]}$ be an $(L+1)K \times K$ variable matrix. Let $Y = (y_{k,j})_{k\in[K],j\in[(L+1)K]}$ be a $K \times (L+1)K$ variable matrix. For every string $\pi \in \Sigma^n$, let $A^{(\pi)}$ be an $(L+1)K \times (L+1)K$ constant matrix.

Definition 3.4 (Iterated Rank Principle [GGRT26])**.** Let L, n, K be positive integers, $\Sigma = \{0, \dots, L\}$, and let $\{A^{(\pi)} : \pi \in \Sigma^n\}$ be a set of $(L+1)K \times (L+1)K$ constant matrices. We define $\mathsf{IRankP}_{L,n,K}(\{A^{(\pi)}\})$ as the system of the following polynomial axioms:

$$\sum_{k\in[K]} x_{i,k}^{(\pi)}y_{k,j} - x_{i,j-\lfloor\frac{j-1}{K}\rfloor K}^{(\pi*\lfloor\frac{j-1}{K}\rfloor)}, \qquad \forall \pi \in \Sigma^{<n}, i, j \in [(L+1)K],$$

$$\sum_{k\in[K]} x_{i,k}^{(\pi)}y_{k,j} - A_{i,j}^{(\pi)}, \qquad \forall \pi \in \Sigma^{n}, i, j \in [(L+1)K].$$

Namely, $\mathsf{IRankP}_{L,n,K}(\{A^{(\pi)}\})$ contains all the degree-2 polynomial equations in the following matrix multiplications:

$$X^{(\pi)}Y = \left[X^{(\pi*0)}\ X^{(\pi*1)}\ \cdots\ X^{(\pi*L)}\right], \qquad \forall \pi \in \Sigma^{<n},$$

$$X^{(\pi)}Y = A^{(\pi)}, \qquad \forall \pi \in \Sigma^{n}.$$

Definition 3.5 (Iterated Rank Principle with extension variables [GGRT26])**.** Let L, n, K be positive integers, $\Sigma = \{0, \dots, L\}$ and let $\{A^{(\pi)} : \pi \in \Sigma^n\}$ be a set of $(L+1)K \times (L+1)K$ constant matrices. We define $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$ as the polynomial system consisting of the axioms of $\mathsf{IRankP}_{L,n,K}(\{A^{(\pi)}\})$ together with the following axioms:

$$z_{i,j,k}^{(\pi)} - x_{i,k}^{(\pi)}y_{k,j}, \qquad \forall \pi \in \Sigma^{\le n}, i, j \in [(L+1)K], k \in [K].$$

We need these extension variables because the axioms of $\mathsf{IRankP}_{L,n,K}(\{A^{(\pi)}\})$ are inner products and thus require additional gates to be computed by an algebraic circuit. Namely, $z_{i,j,k}^{(\pi)}$ correspond to the $\times$ gates computing the products $x_{i,k}^{(\pi)}y_{k,j}$, which we later add by using a $+$ gate. The details of this construction are given in the proof of Theorem 3.7 below.

The monomial size lower bound on $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$ against $\mathsf{PCR}_{\mathbb{F}_2}$ is given in [GGRT26].

Theorem 3.6 ([GGRT26])**.** *Every* $\mathsf{PCR}_{\mathbb{F}_2}$ *refutation of* $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$ *requires* $2^{K^{\Omega(1)}}$ *monomials.*

Theorem 3.7. *There exists a constant $c > 0$ such that given a polynomial f of degree at most ℓ over $\mathbb{F}_2$ in n variables and $s \geq n^c$, every $\mathsf{PCR}_{\mathbb{F}_2}$ refutation of $\mathsf{AlgCircuit}(f, s, \ell)$ requires superpolynomially many monomials.*

Proof. We prove this theorem by construction a reduction from $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$ to $\mathsf{AlgCircuit}(f, s, \ell)$ for some L and K that are polynomial in s and n. The reduction is done by substituting the variables of $\mathsf{AlgCircuit}(f, s, \ell)$ with *constants and variables* of $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$, which does not increase the refutation size. Under this reduction, each axiom of $\mathsf{AlgCircuit}(f, s, \ell)$ is either satisfied or becomes a simple consequence of the axioms of $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$.

The high level idea of the construction is as follows: we explicitly define a small algebraic circuit C that computes the given polynomial f. We proceed by embedding C into $\mathsf{AlgCircuit}(f, s, \ell)$ by fixing all the variables that are responsible for the topology of C (i.e., the variables that encode the type and inputs of each gate in C). This embedding satisfies all the axioms of $\mathsf{AlgCircuit}(f, s, \ell)$ except for those that describe the coefficient vectors of the gates in C. We show that they have simple derivations from the axioms of $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$.

We construct a small circuit $C(\overline{u}) = C(u_1, \ldots, u_n)$ which purportedly computes f and uses variables of $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$ as constants. Let $L \leq \ell$ be the degree of f and let $\Sigma = \{0, \ldots, L\}$. The circuit uses gates $g_x[t, i, k]$ and $g_z[t, i, j, k, d]$, where $t \in \{0, \ldots, n\}$, $i, j \in [(L+1)K]$, $k \in [K]$, and $d \in \Sigma$. We also have a special output gate out. For each $\pi \in \Sigma^{\leq n}$, we write $\overline{u}^{\pi}$ to denote the monomial $u_1^{\pi_1} u_2^{\pi_2} \cdots u_{|\pi|}^{\pi_{|\pi|}}$.

Then for every $\pi \in \Sigma^n$, we define the matrix $A^{(\pi)}$ such that $A^{(\pi)}_{1,1} = \mathrm{coeff}_{\overline{u}^{\pi}(f(\overline{u}))}$ and is zero everywhere else. Assuming that $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$ is satisfied, we require that for each $\pi \in \Sigma^n$,

$$\mathrm{coeff}_{\overline{u}^{\pi}}(\widehat{C}_{\mathsf{out}}) = (X^{(\pi)}Y)_{1,1}. \tag{1}$$

We define $g_z[t, i, j, k, d] = g_x[t, i, k] y_{k,j} u_{t+1}^d$ for every $i, j \in [(L+1)K], k \in [K]$, $d \in \Sigma$ and $t \in \{0, \ldots, n-1\}$ and $g_z[n, 1, 1, k, 0] = g_x[n, 1, k] y_{k,1}$ (the other gates of the form $g_z[n, i, j, k, d]$ with $(i, j, d) \neq (1, 1, 0)$ are not used in the construction). The gates $g_x[t, i, k]$ are thus defined recursively on t, using g_z. For every $i \in [(L+1)K]$ and $k \in [K]$:

$$\begin{aligned} g_x[0, i, k] &= x^{()}_{i,k}, \\ g_x[t, i, k] &= \sum_{d \in \Sigma} \sum_{k' \in [K]} g_z[t-1, i, k + dK, k', d], && \forall t \in [n]. \end{aligned}$$

Here, () denotes the empty string. The output gate of $C(\overline{u})$ is

$$\mathsf{out} = \sum_{k' \in [K]} g_z[n, 1, 1, k', 0]$$

. It is easy to see that $C(\overline{u})$ is an algebraic circuit of size at most $2n(L+1)^3K^3$.

Now we verify that (1) holds for this construction. Before doing this, we start by verifying that for each $t \in \{0, \ldots, n\}$, $i \in [(L+1)K]$ and $k \in [K]$,

$$\widehat{C}_{g_x[t,i,k]} = \sum_{\pi \in \Sigma^{\leq t}} x^{(\pi)}_{i,k} \overline{u}^{\pi}, \tag{2}$$

can be derived using the axioms of $\mathsf{IRankPE}_{L,n,K}(\{A^{\pi}\})$. We prove the statement by induction on t. The base case holds by the definition of $g[0, i, k]$. The inductive step is as follows:

$$\begin{aligned} \widehat{C}_{g_x[t,i,k]} &= \sum_{d \in \Sigma} \sum_{k' \in [K]} g_z[t-1, i, k + dK, k', d] && \text{(by definition)} \\ &= \sum_{d \in \Sigma} \left(\sum_{k' \in [K]} g_x[t-1, i, k'] y_{k', k+dK} \right) u_t^d && \text{(expanding } g_z\text{)} \end{aligned}$$

$$= \sum_{d\in\Sigma}\left(\sum_{k'\in[K]}\left(\sum_{\pi\in\Sigma^{\le t-1}} x^{(\pi)}_{i,k'}\overline{u}^{\pi}\right) y_{k',k+dK}\right) u_t^d \qquad \text{(by the induction hypothesis)}$$

$$= \sum_{d\in\Sigma}\left(\sum_{\pi\in\Sigma^{\le t-1}}\left(\sum_{k'\in[K]} x^{(\pi)}_{i,k'} y_{k',k+dK}\right)\overline{u}^{\pi}\right) u_t^d \qquad \text{(switching the summation order)}$$

$$= \sum_{d\in\Sigma}\left(\sum_{\pi\in\Sigma^{\le t-1}} x^{(\pi*d)}_{i,k}\overline{u}^{\pi}\right) u_t^d \qquad \text{(by the axioms of } \mathsf{IRankPE}_{L,n,K}(\{A^{\pi}\}))$$

$$= \sum_{\pi\in\Sigma^{\le t}} x^{(\pi)}_{i,k}\overline{u}^{\pi}. \qquad \text{(combining } \overline{u}^{\pi} \text{ and } u_t^d)$$

Similarly, we show that

$$\widehat{C}_{\mathsf{out}} = \sum_{\pi\in\Sigma^{\le n}} A^{(\pi)}_{1,1}\overline{u}^{\pi}$$

, which implies (1) due to the definition of $A^{(\pi)}$.

$$\widehat{C}_{\mathsf{out}} = \sum_{k'\in[K]} g_z[n,1,1,k',0] \qquad \text{(by definition)}$$

$$= \sum_{k'\in[K]} g_x[n,1,k']y_{k',1} \qquad \text{(expanding } g_z)$$

$$= \sum_{k'\in[K]}\left(\sum_{\pi\in\Sigma^{\le n}} x^{(\pi)}_{1,k'}\overline{u}^{\pi}\right) y_{k',1} \qquad \text{(by (2))}$$

$$= \sum_{\pi\in\Sigma^{\le n}}\left(\sum_{k'\in[K]} x^{(\pi)}_{1,k'} y_{k',1}\right)\overline{u}^{\pi} \qquad \text{(switching the summation order)}$$

$$= \sum_{\pi\in\Sigma^{\le n}} A^{(\pi)}_{1,1}\overline{u}^{\pi}. \qquad \text{(by the axioms of } \mathsf{IRankPE}_{L,n,K}(\{A^{\pi}\}))$$

Now, we present a substitution ρ, which embeds the circuit C into $\mathsf{AlgCircuit}(f,s,\ell)$. We begin by encoding the topology of the circuit by assigning all the Type, $\mathsf{InputType}$, $\mathsf{InputType}'$, $\mathsf{InputType}''$, $\mathsf{InputVar}$ and $\mathsf{InputNode}$ variables to the appropriate constants. Slightly abusing the notation, we proceed by setting for each $t\in\{0,\dots,n\}, i\in[(L+1)K], k\in[K], \pi\in\Sigma^{\le t}$, $d\in\Sigma$,

$$\rho(\alpha_{\overline{u}^{\pi},g_x[t,i,k]}) = x^{(\pi)}_{i,k}$$

and $\rho(\alpha_{M,g_x[t,i,k]}) = 0$ for the remaining monomials M. The values of $g_z[t,i,j,k]$ are defined similarly. Then the inputs of $g_x[t,i,k]$ are defined as the appropriate gates $g_z[t-1,i,k+dK,k',d]$ to satisfy the definition of C. The inputs of $g_z[t,i,j,k,d]$ are similarly defined as the gate $g_x[t,i,k]$, followed by d variables u_{t+1} and a constant $y_{k,j}$.

Observe that such a substitution replaces each variable with either a constant or another variable (possibly a negation). This preserves the PCR size of the refutation. It follows from the definition that the axioms of $\mathsf{AlgCircuit}(f,s,\ell)\restriction\rho$ can be easily derived from the axioms of $\mathsf{IRankPE}_{L,n,K}(\{A^{(\pi)}\})$.

Now suppose that $\mathsf{AlgCircuit}(f,s,\ell))$ has a PCR refutation of polynomial size, specifically,

$$\mathrm{poly}(s,n,N^{\ell}) \le \exp((\ell n)^{O(1)})$$

. Then, $\mathsf{IRankPE}_{L,n,K}(\{A_{\pi}\})$ has a PCR refutation of size $\exp((\ell n)^{O(1)})$. By taking K to be sufficiently large (say, $(\ell n)^{\Omega(1)}$), this contradicts Theorem 3.6 since such a refutation must have size $\exp(K)$. □

Similarly to $\mathsf{AlgCircuit}(f, s, \ell)$, we can define $\mathsf{AlgCircuit}(f, s, \ell, \Delta)$ stating that an n-variable polynomial f has a depth-Δ algebraic circuit of size s and syntactic-degree ℓ. We can apply a similar idea to get a lower bound on the size of PCR refutations of $\mathsf{AlgCircuit}(f, s, \ell, \Delta)$ provided the rank principle does not hold.

Theorem 3.8. *For every constant $c > 0$, given a polynomial f of degree at most ℓ over $\mathbb{F}_2$ in n variables and $s \geq N^{1/c}$, every $\mathsf{PCR}_{\mathbb{F}_2}$ refutation of $\mathsf{AlgCircuit}(f, s, \ell, O(c))$ requires superpolynomially many monomials.*

Proof sketch. We construct the circuit C similarly to the one from the proof of Theorem 3.7 but instead of multiplying the $\overline{u}$-variables one-by-one, we multiply them by all monomials of degree at most ℓ in the subsequent n/c variables. This increases the size of the circuit since we are forced to choose L to be the number of such monomials, which is approximately $N^{1/c}$. However, this way, C only requires depth $O(c)$. □

Remark 3.9. Observe that the above circuit constructions do not use the whole matrices $A^{(\pi)}$, but only their $(1,1)$th entry. We can construct some slightly more efficient circuits by using more entries of $A^{(\pi)}$ to store the coefficients by computing the polynomials

$$\sum_{\pi \in \Sigma^{\leq n-2}, d, d' \in \Sigma} A^{(\pi)}_{d,d'} u^{(\pi)} u^d_{n-1} u^{d'}_n .$$

This can be done by introducing gates $\alpha[k]$ computing the polynomials

$$\sum_{d \in \Sigma} \sum_{\pi \in \Sigma^{\leq n-2}} x^{(\pi)}_{d,k} u^{(\pi)} u^d_{n-1}$$

and gates $\beta[d' + dK]$ computing $y_{d'+dK} u^{d'}_n$. Then, the output gate can be defined as the inner product of α and β.

A similar idea can be applied to the proof for $\mathsf{AlgCircuit}(f, s, \ell, O(c))$.

Remark 3.10. Note that we can also view $\mathsf{AlgCircuit}(f, s, \ell)$ as a **proof complexity generator** in the algebraic setting: treating the coefficient vector of f as an output vector $\overline{b} \in \mathbb{F}^N$ (where N is the number of monomials of degree at most ℓ over n variables) of a function $\tau(s, \ell)$, we can view $\mathsf{AlgCircuit}(f, s, \ell)$ as an encoding of the statement $\tau(s, \ell) = \overline{b}$. Thus, Theorem 3.7 shows that $\mathsf{AlgCircuit}(f, s, \ell)$ is a proof complexity generator against PCR.

Similarly, $\mathsf{AlgCircuit}(f, s, \ell, \Delta)$ can be viewed as $\tau(s, \ell, \Delta) = \overline{b}$, where $\overline{b} \in \mathbb{F}^N$ and N is the number of monomials of degree at most ℓ over n variables, and is a proof complexity generator against PCR.

4 Hardness of Bounded-Depth Algebraic Circuit Lower Bounds from Hardness of Weak Tensor Rank Principles

In this section, we need a generalisation of the rank principle, which we call the *tensor rank principle*.

Definition 4.1 (Tensor Rank Principle)**.** Let $\mathbb{F}$ be a field and m, n two positive integers with $m > n$. We denote $\mathsf{TRankP}^m_n(r, A)$ the system of degree-r polynomial equations stating that the tensor rank of an r-tensor $A \in \mathbb{F}^{\overbrace{m \times \cdots \times m}^{r \text{ times}}}$ is at most n over $\mathbb{F}$. More precisely, $\mathsf{TRankP}^m_n(r, A)$ is defined over rmn variables arranged into rn vectors $\mathbf{x}_{j,k} \in \mathbb{F}^m : j \in [r], k \in [n]$.

For every $i_1, \ldots, i_r \in [m]$ (which are not necessarily distinct), there is an equation in $\mathsf{TRankP}_n^m(r, A)$ stating that the $(i_1, \ldots, i_r)$th entry in the summation of the tensor product $\bigotimes_{j=1}^r \mathbf{x}_{j,k}$ is equal to $A_{i_1,\ldots,i_r}$. That is, $\mathsf{TRankP}_n^m(r, A)$ consists of the polynomials

$$\sum_{k\in[n]} \prod_{j\in[r]} x_{j,k,i_j} - A_{i_1,\ldots,i_r}, \qquad i_1, \ldots, i_r \in [m], \tag{3}$$

where x_{j,k,i_j} denotes the (i_j)th entry of the vector $\mathbf{x}_{j,k}$.

Namely, $\mathsf{TRankP}_n^m(r, A)$ is the set of polynomial equations stating that

$$\sum_{k\in[n]} \bigotimes_{j=1}^{r} \mathbf{x}_{j,k} = A.$$

By definition, $\sum_{k\in[n]} \bigotimes_{j=1}^r \mathbf{x}_{j,k}$ is the tensor rank decomposition of rank at most n. Thus, when the tensor rank of A exceeds n, $\mathsf{TRankP}_n^m(r, A)$ is unsatisfiable. This principle can be viewed as a generalisation of the rank principle, and coincides with it when $r = 2$.

Theorem 4.2. *Let m, n, r be positive integers with $m > n$. Let $J \subseteq [m]$ with $|J| > n$ and let A be the r-tensor defined as*

$$A_{i_1,\ldots,i_r} = \begin{cases} 1 & \text{if } i_1 = \cdots = i_r \in J, \\ 0 & \text{otherwise.} \end{cases}$$

Then every $\mathsf{PCR}_{\mathbb{F}_2}$ refutation of $\mathsf{TRankP}_n^m(r, A)$ requires $2^{\Omega(n)}$ monomials.

Proof. Without loss of generality assume that $A = I_m$ (and thus $J = [m]$): we can set all $x_{j,k,i}$ to 0 for all $j \in [r]$, $k \in [n]$ and $i \notin J$. This reduces $\mathsf{TRankP}_n^m(r, A)$ to $\mathsf{TRankP}_n^{|J|}(r, I_{|J|})$.

The idea of the proof is analogous to the proof of the lower bound for RankP in [GGRT26].

We say that a term t *mentions* an index $k \in [n]$ if there exist $j \in [r]$ and $i \in [m]$ such that $x_{j,k,i}$ or $\overline{x}_{j,k,i}$ appears in t. We start by proving a lower bound on the number of different indices $k \in [n]$ each term mentions by reducing PHP_n^m to $\mathsf{TRankP}_n^m(r, A)$. The axioms of PHP_n^m are the *pigeon axioms*

$$\sum_{k\in[n]} x_{i,k} - 1, \qquad \forall i \in [m],$$

and the *hole axioms*

$$x_{i,k}x_{j,k}, \qquad \forall i \neq j \in [m], k \in [n].$$

It follows from [Raz98, IPS99] that any $\mathsf{PCR}_{\mathbb{F}_2}$ refutation of PHP_n^m must have a term that mentions at least $n/2+1$ different holes $k \in [n]$ (in fact, it must contain a term that correspond to a *matching* of this size).

Let $X = (x_{i,k})_{i\in[m],k\in[n]}$ be fresh variables and define the restriction ρ as

$$\rho(x_{j,k,i}) = x_{i,k}, \qquad \forall j \in [d], k \in [n], i \in [m].$$

Under ρ, the axioms of $\mathsf{TRankP}_n^m(r, A)$ become[5]

$$\sum_{k\in[n]} x_{i,k} - 1, \tag{4}$$

when $i_1 = \cdots = i_r = i$ and

$$\sum_{k\in[n]} \prod_{j\in[r]} x_{i_j,k}, \tag{5}$$

[5]Recall that proof lines are *multilinear* polynomials.

where at least two i_j and $i_{j'}$ are different. Observe that (4) is precisely the pigeon axiom of PHP^m_n and (5) can be derived easily from the hole axioms $x_{i_j,k}x_{i_{j'},k}$. Thus, every $\mathsf{PCR}_{\mathbb{F}_2}$ refutation of $\mathsf{TRankP}^m_n(r, A)$ must contain a term involving at least $n/2+1$ variables $x_{j,k,i}$ (or their negations) with different k.

Now let σ be a partial restriction defined as follows. Choose a random subset $K \subseteq [n-1]$ of size $n/2-1$ uniformly and for each $k \in K$ independently choose $c_k \in \{0,1\}$ at random. Define $c_n = 1$ if $\sum_{k\in K} c_k$ is odd and $c_n = 0$ otherwise. This guarantees that $\sum_{k\in K\cup\{n\}} c_k$ is even. Then assign

$$\sigma(x_{j,k,i}) = \begin{cases} c_k & \text{if } k \in K \cup \{n\}, \\ * & \text{if } k \notin K \cup \{n\}. \end{cases}$$

Under σ, $\mathsf{TRankP}^m_n(r, A) \restriction \sigma$ becomes $\mathsf{TRankP}^m_{n/2}(r, A)$ (up to the variable renaming). Moreover, each term that mentions at least $n/2+1$ different $k \in [n]$, becomes zero with probability $1 - \exp(-\Omega(n))$ by the Chernoff bound. Applying the union bound, we can eliminate all such terms in a proof of size at most $\exp(\epsilon n)$ for a sufficiently small constant $\epsilon > 0$, contradicting the above degree bound. □

Theorem 4.3. *Let $n \geq 1$ and let $f \in \mathbb{F}_2[u_1, \ldots, u_n]$ have degree at most $\ell \geq 2$. Set*

$$D = n\ell, \qquad N = \binom{n+\ell}{\ell}, \qquad R = \binom{n+D}{D}, \qquad K = \lfloor \sqrt{s} \rfloor.$$

Assume $K < N$, $\Delta > 4$, and $s \geq 4n^2(\ell+1)^4$.

Let A be the diagonal order-n tensor of dimension N whose $(M, \ldots, M)$th entry is $\mathrm{coeff}_M(f)$, with all other entries zero.

If $\mathsf{AlgCircuit}(f, s, D, \Delta)$ admits a Δ'-IPS refutation of size S over $\mathbb{F}_2$, then $\mathsf{TRankP}^N_K(n, A)$ admits a $(\Delta'+5)$-IPS refutation over $\mathbb{F}_2$ of size

$$S + \mathrm{poly}(s, n, \ell, K, N^n, R^D).$$

The proof of this theorem is analogous to the proof of Theorem 3.7. We construct a circuit C parametrised by the variables of $\mathsf{TRankP}^N_K(n, A)$, with depth at most 4, size at most s, and syntactic degree at most $D = n\ell$. We then embed this circuit into $\mathsf{AlgCircuit}(f, s, D, \Delta)$ and construct bounded-depth IPS derivations of the substituted circuit axioms from the axioms of $\mathsf{TRankP}^N_K(n, A)$. Consequently, a short IPS refutation of $\mathsf{AlgCircuit}(f, s, D, \Delta)$ yields a short IPS refutation of $\mathsf{TRankP}^N_K(n, A)$.

Proof. We show how to reduce the bounded-depth algebraic circuit upper bound formulas to $\mathsf{TRankP}^N_K(n, A)$, where $T_f \coloneqq \{M : M \text{ is a monomial in } f \text{ with a non-zero coefficient}\}$ and K is bounded by $\sqrt{s}$.

We use the notation $\deg_{u_j}(M)$ to denote the individual degree of u_j in the monomial M.

Consider the depth-4 algebraic circuit

$$C(u_1, \ldots, u_n) \coloneqq \sum_{k\in[K]} \prod_{j\in[n]} \sum_{d\in\{0,1,\ldots,\ell\}} \alpha_{j,k,d} u_j^d$$

where $\alpha_{j,k,d}$ are placeholder variables defined as

$$\alpha_{j,k,d} \coloneqq \sum_{\substack{M\in T_f \\ \deg_{u_j}(M)=d}} x_{j,k,M}.$$

The syntactic degree of C is at most $D = n\ell$.

For every monomial $M = u_1^{d_1} \cdots u_n^{d_n}$ of degree at most D with $d_j \le \ell$ for all j, let $X_{k,M}$ denote the sum of all off-diagonal terms in the expansion of $\prod_{j\in[n]} \alpha_{j,k,d_j}$. More precisely,

$$X_{k,M} \coloneqq \sum_{\substack{M_1,\dots,M_n \in T_f \\ \deg_{u_j}(M_j)=d_j \ \forall j \\ (M_1,\dots,M_n) \text{ not diagonal}}} \prod_{j\in[n]} x_{j,k,M_j}.$$

The off-diagonal axioms of $\mathsf{TRankP}_K^N(n, A)$ imply

$$\sum_{k\in[K]} X_{k,M} \equiv 0,$$

where $\equiv$ denotes congruence modulo the ideal generated by the tensor-rank axioms.

For every $M = u_1^{d_1} \cdots u_n^{d_n} \in T_f$, we have

$$\begin{aligned}
\operatorname{coeff}_M(\widehat{C}) &= \operatorname{coeff}_M\left(\sum_{k\in[K]} \prod_{j\in[n]} \sum_{d=0}^{\ell} \alpha_{j,k,d} u_j^d\right) && \text{(by definition)} \\
&= \sum_{k\in[K]} \operatorname{coeff}_M\left(\prod_{j\in[n]} \sum_{d=0}^{\ell} \alpha_{j,k,d} u_j^d\right) && \text{(since coefficients add termwise)} \\
&= \sum_{k\in[K]} \prod_{j\in[n]} \alpha_{j,k,d_j} && \text{(computing the coefficient for } M\text{)} \\
&= \sum_{k\in[K]} \left(\prod_{j\in[n]} x_{j,k,M} + X_{k,M}\right) && \text{(expanding the products)} \\
&\equiv \sum_{k\in[K]} \prod_{j\in[n]} x_{j,k,M} && \text{(by the off-diagonal tensor axioms)} \\
&\equiv A_{M,\dots,M} && \text{(by the diagonal tensor axiom)} \\
&= \operatorname{coeff}_M(f). && \text{(by the definition of } A\text{)}
\end{aligned}$$

For a monomial $M \notin T_f$ of degree at most D with $d_j \le \ell$ for all j, we have

$$\begin{aligned}
\operatorname{coeff}_M(\widehat{C}) &= \sum_{k\in[K]} \operatorname{coeff}_M\left(\prod_{j\in[n]} \sum_{d=0}^{\ell} \alpha_{j,k,d} u_j^d\right) && \text{(since coefficients add termwise)} \\
&= \sum_{k\in[K]} X_{k,M} && \text{(computing the coefficient for } M\text{)} \\
&\equiv 0 && \text{(by the off-diagonal tensor axioms)} \\
&= \operatorname{coeff}_M(f).
\end{aligned}$$

If $d_j > \ell$ for some j, then $\operatorname{coeff}_M(\widehat{C}) = \operatorname{coeff}_M(f) = 0$ identically.

Consequently, for every monomial M of degree at most $D = n\ell$, the output-coefficient equation

$$\operatorname{coeff}_M(\widehat{C}) - \operatorname{coeff}_M(f) = 0$$

has an IPS derivation from the axioms of $\mathsf{TRankP}_K^N(n, A)$.

We construct a substitution ρ that embeds C into $\mathsf{AlgCircuit}(f, s, n\ell, \Delta)$. Note that C is a depth-4 algebraic circuit having size at most $2Kn(\ell+1)^2$ and syntactic degree at most $n\ell$. The formal variables $\alpha_{j,k,d}$ are replaced with $\sum_{\substack{M\in T_f, \\ \deg_{u_j}(M)=d}} x_{j,k,M}$ of size at most N.

Under this substitution, every axiom in $\mathsf{AlgCircuit}(f, s, n\ell, \Delta)$, where $s \geq 2Kn(\ell+1)^2$, is transformed into a polynomial derivable from $\mathsf{TRankP}^N_K(n, A)$ with total size at most

$$\mathrm{poly}(s, n, \ell, K, N^n, R^{n\ell}), \qquad R = \binom{n + n\ell}{n\ell},$$

and constant depth. In particular, the high-degree multiplication axioms vanish identically, and the output-coefficient axioms follow from the tensor-rank axioms. Consequently, if $\mathsf{AlgCircuit}(f, s, n\ell, \Delta)$ admits a Δ'-IPS refutation of size S, then $\mathsf{TRankP}^N_K(n, A)$ admits a $(\Delta'+5)$-IPS refutation of size

$$S + \mathrm{poly}(s, n, \ell, K, N^n, R^{n\ell}).$$

Finally, since $s \geq 4n^2(\ell+1)^4$, we may set $K = \lfloor \sqrt{s} \rfloor$, which guarantees $2Kn(\ell+1)^2 \leq s$. □

Corollary 4.4. *Consider a family of instances satisfying the hypotheses of Theorem 4.3, with $K \to \infty$. Let*

$$D = n\ell, \qquad R = \binom{n + D}{D}.$$

Suppose that, for some fixed $\varepsilon > 0$ and constant $d > 5$, the corresponding systems $\mathsf{TRankP}^N_K(n, A)$ require d-IPS refutations over $\mathbb{F}_2$ of size at least 2^{K^ε}, and that

$$\log(sn) + n \log N + D \log R = o(K^\varepsilon).$$

Then $\mathsf{AlgCircuit}(f, s, D, \Delta)$ requires $(d-5)$-IPS refutations over $\mathbb{F}_2$ of size $|\mathsf{AlgCircuit}(f, s, D, \Delta)|^{\omega(1)}$.

Proof. The encoding has size polynomial in s, n, R^D. Under the stated growth condition, both every fixed polynomial in this encoding size and the additive overhead in Theorem 4.3 are $2^{o(K^\varepsilon)}$. A polynomial-size $(d-5)$-IPS refutation would therefore give a d-IPS refutation of the tensor system of size less than 2^{K^ε}, a contradiction. □

Remark 4.5. The tensor A in Theorem 4.3 is diagonal. Over $\mathbb{F}_2$, its restriction to the coordinates in the support of f is a unit tensor. The assumption in Corollary 4.4 concerns these diagonal tensor rank instances. For order 2, the tensor rank principle coincides with the matrix rank principle. Establishing constant-depth IPS lower bounds for the diagonal tensor rank instances remains the hypothesis of our conditional result.

5 Short NC^2-Frege Proofs for Constant-Depth Algebraic Circuit Lower Bounds

In this section, we show that NC^2-Frege (see [HT15] for the definition) admits short proofs of the constant-depth algebraic circuit lower bounds for the Permanent and Determinant established in [LST25, For24].

For this purpose, we use a formalisation in VNC^2, which is a first-order theory whose (low logical complexity) theorems translate to polynomial-size proofs in NC^2-Frege. We shall demonstrate a formalisation of the proof of the superpolynomial lower bound of the Permanent polynomial against constant-depth algebraic circuits over any field. This is an adaptation of the proof of [LST25] and [For24]. The work in [LST25] demonstrated a superpolynomial lower bound against constant-depth algebraic circuits over fields of large characteristic. Forbes extended this result to any field by replacing the original set-multilinearization argument in [LST25], which works only over fields of large characteristic, with a new set-multilinearization argument over any field.

In the previous sections, we presented lower-bound (or unprovability) results for constant-depth algebraic circuit lower bound statements. In this section, we show an *upper bound* on proving constant-depth algebraic circuit lower bounds.

5.1 Preliminaries for Bounded Arithmetic

For the convenience of the reader, we provide nearly full details of the theory VNC^2 in which we work. We refer the reader to [CN10] for more information.

5.1.1 The theory V^0

Definition 5.1 (Language of two-sorted arithmetic $\mathcal{L}_A^2$)**.** The language of two-sorted arithmetic, denoted $\mathcal{L}_A^2$, consists of the following relation, function and constant symbols:

$$\{+, \cdot, \leq, 0, 1, |\cdot|, =_1, =_2, \in\}.$$

A number quantifier is said to be *bounded* if it is of the form $\exists x(x \leq t \wedge \cdots)$ or $\forall x(x \leq t \rightarrow \cdots)$ for some *number term* t that does not contain x. We abbreviate $\exists x(x \leq t \wedge \cdots)$ and $\forall x(x \leq t \rightarrow \cdots)$ by $\exists x \leq t$ and $\forall x \leq t$, respectively. A string quantifier is said to be *bounded* if it is of the form $\exists X(|X| \leq t \wedge \cdots)$ or $\forall X(|X| \leq t \rightarrow \cdots)$ for some *number term* t that does not contain X. We abbreviate $\exists X(|X| \leq t \wedge \cdots)$ and $\forall X(|X| \leq t \rightarrow \cdots)$ by $\exists X \leq t$ and $\forall X \leq t$, respectively.

A formula is in the class of formulas Σ_0^B or Π_0^B if it uses *no string quantifiers* and all number quantifiers are bounded. A formula is in Σ_{i+1}^B or Π_{i+1}^B if it is of the form $\exists X_1 \leq t_1 \cdots \exists X_m \leq t_m \psi$ or $\forall X_1 \leq t_1 \cdots \forall X_m \leq t_m \psi$, where $\psi \in \Pi_i^B$ and $\psi \in \Sigma_i^B$, respectively, and none of $X_1, \ldots, X_m$ occurs in any of $t_1, \ldots, t_m$.

A $\Sigma_1^1(\mathcal{L})$-formula is one of the form $\exists \vec{X} \varphi$, where $\vec{X}$ is a vector of zero or more string variables, and φ is a $\Sigma_0^B(\mathcal{L})$-formula.

We write $T(t)$ to abbreviate $t \in T$, for a number term t and a string term T. For a formula ψ, we write $\psi(a/x)$ to denote the *substitution instance* of ψ in which every free occurrence of the variable x is replaced by the term a.

A finite set N of natural numbers represents the finite binary string $S_N = S_N^0 \ldots S_N^{|N|-1}$ such that $S_N^i = 1$ if and only if $i \in N$.

Definition 5.2 (Comprehension Axiom)**.** If Φ is a set of formulas, then the *comprehension axiom scheme* for Φ, denoted by Φ-**COMP**, is the set of all formulas

$$\exists X \leq y \, \forall z < y \, (z \in X \leftrightarrow \varphi(z)),$$

where φ is a formula in Φ and X does not occur free in φ.

Definition 5.3 (V^0)**.** The theory V^0 has the vocabulary $\mathcal{L}_A^2$ and is axiomatized by 2-Basic and Σ_0^B-**COMP**. The axioms of 2-Basic are listed in Table 3.

B1. $x + 1 \neq 0$	**B7**. $(x \leq y \wedge y \leq x) \supset x = y$
B2. $x + 1 = y + 1 \supset x = y$	**B8**. $x \leq x + y$
B3. $x + 0 = x$	**B9**. $0 \leq x$
B4. $x + (y + 1) = (x + y) + 1$	**B10**. $x \leq y \vee y \leq x$
B5. $x \cdot 0 = 0$	**B11**. $x \leq y \leftrightarrow x < y + 1$
B6. $x \cdot (y + 1) = (x \cdot y) + x$	**B12**. $x \neq 0 \supset \exists y \leq x \, (y + 1 = x)$
L1. $X(y) \supset y < \|X\|$	**L2**. $y + 1 = \|X\| \supset X(y)$
SE. $(\, \|X\| = \|Y\| \ \wedge \ \forall i < \|X\| \, (\, X(i) \leftrightarrow Y(i) \,) \,) \ \supset \ X = Y$	

Table 3: Axioms of 2-Basic.

Definition 5.4 (Number Induction Axiom)**.** If Φ is a set of two-sorted formulas, then Φ-**IND** axioms are the formulas

$$\varphi(0) \wedge \forall x\ (\varphi(x) \rightarrow \varphi(x+1)) \rightarrow \forall z \varphi(z)$$

where φ is a formula in Φ.

Definition 5.5 (Number Minimization and Maximization Axioms)**.** The number minimization axioms (or *least number principle* axioms) for a set Φ of two-sorted formulas are denoted Φ-**MIN** and consist of the formulas

$$\varphi(y) \rightarrow \exists x \leq y(\varphi(x) \wedge \neg \exists z < x \varphi(z))$$

where φ is a formula in Φ. Similarly, the number maximization axioms for Φ are denoted Φ-**MAX** and consist of the formulas

$$\varphi(0) \rightarrow \exists x \leq y(\varphi(x) \wedge \neg \exists z \leq y(x < z \wedge \varphi(z)))$$

where φ is a formula in Φ.

Lemma 5.6. *Let $\mathcal{T}$ be an extension of V^0 and Φ be a set of formulas in $\mathcal{T}$. Suppose that $\mathcal{T}$ proves the Φ-**COMP** axiom scheme. Then $\mathcal{T}$ also proves the Φ-**IND** axiom scheme, the Φ-**MIN** axiom scheme, and the Φ-**MAX** axiom scheme.*

Proposition 5.7. *The theory V^0 proves Σ_0^B-**IND**, Σ_0^B-**MIN**, and Σ_0^B-**MAX**.*

Definition 5.8 (Two-Sorted Polynomially Bounded Functions)**.** A *number function* is a two-sorted function whose output belongs to the number sort. A *string function* is a two-sorted function whose output belongs to the string sort.

A number function f is *polynomially bounded* (or *p-bounded*) if there is a polynomial p such that $f(\vec{x}, \vec{Y}) \leq p(\vec{x}, |\vec{Y}|)$. A string function F is p-bounded if $|F(\vec{x}, \vec{Y})| \leq p(\vec{x}, |\vec{Y}|)$ for some polynomial p.

Definition 5.9 (Graph, Bit Graph)**.** The *graph* G_f of a number function $f(\vec{x}, \vec{Y})$ is defined by

$$G_f(z, \vec{x}, \vec{Y}) \leftrightarrow z = f(\vec{x}, \vec{Y}).$$

The *bit graph* B_F of a string function $F(\vec{x}, \vec{Y})$ is defined by

$$B_F(i, \vec{x}, \vec{Y}) \leftrightarrow F(\vec{x}, \vec{Y})(i).$$

Later, we shall introduce bit-definability of new string function symbols based on the notion of bit graph.

Definition 5.10 (Function class)**.** If C is a two-sorted complexity class of relations, then the corresponding function class FC consists of all p-bounded number functions whose graphs are in C, together with all p-bounded string functions whose bit graphs are in C.

Saying that a graph G_f (respectively, a bit graph B_F) is in C means that there exists a family of circuits in C which, given z (respectively, i), the tuple $\vec{x}$ in unary, and the strings $\vec{Y}$, decides whether $(z, \vec{x}, \vec{Y})$ is in G_f (respectively, $(i, \vec{x}, \vec{Y})$ is in B_F).

Note that C is a uniform complexity class.

Definition 5.11 (FAC^0)**.** A string (respectively, number) function is in FAC^0 if it is polynomially bounded and its bit graph (respectively, graph) is definable by a Σ_0^B-formula in the language $\mathcal{L}_A^2$.

Definition 5.12 (Two-sorted definability in a theory $\mathcal{T}$)**.** Let $\mathcal{T}$ be a theory in a language $\mathcal{L} \supseteq \mathcal{L}_A^2$ and let Φ be a set of formulas in the language $\mathcal{L}$. A *number function* f is Φ-definable in the theory $\mathcal{T}$ iff there is a formula $\varphi(\vec{x}, y, \vec{X})$ in Φ such that

$$\mathcal{T} \vdash \forall \vec{x} \forall \vec{X} \exists! y \varphi(\vec{x}, y, \vec{X}), \tag{6}$$

and it holds in the standard two-sorted model $\mathcal{N}_2$ (whose number sort is $\mathbb{N}$ and the set sort is the set of finite subsets of $\mathbb{N}$) that

$$y = f(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, y, \vec{X}). \tag{7}$$

A *string function* F is Φ-definable in the theory $\mathcal{T}$ iff there is a formula $\varphi(\vec{x}, \vec{X}, Y)$ in Φ such that

$$\mathcal{T} \vdash \forall \vec{x} \forall \vec{X} \exists! Y \varphi(\vec{x}, \vec{X}, Y) \tag{8}$$

and it holds in the standard two-sorted model $\mathcal{N}_2$ that

$$Y = F(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, \vec{X}, Y). \tag{9}$$

The formulas (7) and (9) are the *defining axioms* for f and F, respectively.

For string functions, $\varphi(\vec{x}, \vec{X}, Y)$ defines the entire output string: it holds exactly when $Y = F(\vec{x}, \vec{X})$. A bit definition instead specifies the individual output bits.

Definition 5.13 (Conservative extension of a theory)**.** Let $\mathcal{T}$ be a theory in a language $\mathcal{L}$. We say that a theory $\mathcal{T}' \supseteq \mathcal{T}$ in a language $\mathcal{L}' \supseteq \mathcal{L}$ is *conservative* over $\mathcal{T}$ if every $\mathcal{L}$ formula provable in $\mathcal{T}'$ is also provable in $\mathcal{T}$.

We may expand the language $\mathcal{L}$ and a theory $\mathcal{T}$ over $\mathcal{L}$. If f is definable in $\mathcal{T}$, in particular if $\mathcal{T}$ proves existence and uniqueness of the value satisfying the defining formula, then adjoining a symbol for f together with its defining axiom gives a conservative extension. The resulting extension, denoted $\mathcal{T}(f)$ for a function f (and similarly for relations), is a conservative extension of $\mathcal{T}$. However, such an extension *does not guarantee* that one can use the new function symbols in the *comprehension* (and *induction*) axiom schemes.

Definition 5.14 (Provably Total Function)**.** A function (either a number function or a string function) is said to be *provably total* in a theory $\mathcal{T}$ iff it is Σ_1^1-definable in $\mathcal{T}$.

We now state an important corollary, which follows from the two-sorted version of Parikh's theorem.

Definition 5.15 (Polynomially Bounded Theory)**.** Let $\mathcal{T}$ be a two-sorted theory over a vocabulary $\mathcal{L} \supseteq \mathcal{L}_A^2$. A number function symbol $f(\vec{x}, \vec{X})$ in $\mathcal{L}$ is said to be *polynomially bounded in* $\mathcal{T}$ if there is an $\mathcal{L}_A^2$-number term $t(\vec{x}, |\vec{X}|)$ such that

$$\mathcal{T} \vdash f(\vec{x}, \vec{X}) \leq t(\vec{x}, |\vec{X}|).$$

Similarly, a string function symbol $F(\vec{x}, \vec{X})$ in $\mathcal{L}$ is said to be *polynomially bounded in* $\mathcal{T}$ if there is an $\mathcal{L}_A^2$-number term $t(\vec{x}, |\vec{X}|)$ such that

$$\mathcal{T} \vdash |F(\vec{x}, \vec{X})| \leq t(\vec{x}, |\vec{X}|).$$

We say that $\mathcal{T}$ is a *polynomially bounded theory* if:

1. $\mathcal{T}$ extends V^0;
2. $\mathcal{T}$ can be axiomatized by a set of bounded formulas; and
3. every number and string function symbol in the vocabulary $\mathcal{L}$ is polynomially bounded in $\mathcal{T}$.

The following is a consequence of the two-sorted version of Parikh's theorem.

Corollary 5.16. *Let $\mathcal{T}$ be a polynomially bounded theory. Then every function provably total in $\mathcal{T}$ is polynomially bounded. Moreover, a function is provably total in $\mathcal{T}$ if and only if it is Σ^B_1-definable in $\mathcal{T}$.*

Definition 5.17 (Representable/Definable Relations)**.** Let $\mathcal{L} \supseteq \mathcal{L}^2_A$ be a two-sorted vocabulary, and let φ be an $\mathcal{L}$-formula. Then we say that $\varphi(\vec{x}, \vec{X})$ *represents* (or *defines*) a relation $R(\vec{x}, \vec{X})$ if

$$R(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, \vec{X}) \tag{10}$$

holds in the standard two-sorted model $\mathcal{N}_2$. If Φ is a set of $\mathcal{L}$-formulas, then we say that $R(\vec{x}, \vec{X})$ is Φ-*representable* (or Φ-*definable*) if it is represented by some $\varphi \in \Phi$.

Definition 5.18 ($\Sigma^B_0(\Phi)$)**.** Let Φ be a set of formulas in a two-sorted language $\mathcal{L} \supseteq \mathcal{L}^2_A$. We denote by $\Sigma^B_0(\Phi)$ the smallest class of formulas containing $\Phi \cup \Sigma^B_0(\mathcal{L})$ and closed under Boolean connectives and bounded number quantification. That is, $\Sigma^B_0(\Phi)$ is closed under $\neg$, $\wedge$, $\vee$, and quantifiers of the form

$$\exists x \leq t \qquad \text{and} \qquad \forall x \leq t,$$

where t is a number term not containing x.

Definition 5.19 (Bit-definable Function)**.** Let Φ be a set of $\mathcal{L}$-formulas, where $\mathcal{L} \supseteq \mathcal{L}^2_A$. We say that a string function symbol $F(\vec{x}, \vec{Y})$ not in $\mathcal{L}$ is Φ-*bit-definable from* $\mathcal{L}$ if there is a formula $\varphi(i, \vec{x}, \vec{Y}) \in \Phi$ and an $\mathcal{L}^2_A$-number term $t(\vec{x}, |\vec{Y}|)$ such that the bit graph of F satisfies

$$F(\vec{x}, \vec{Y})(i) \leftrightarrow (i < t(\vec{x}, |\vec{Y}|) \wedge \varphi(i, \vec{x}, \vec{Y})). \tag{11}$$

We call the formula on the right-hand side of (11) a *bit-defining axiom*, or a *bit definition*, for F.

One crucial point is that adding a Φ-bit-definable string function F together with its bit-defining axiom need not yield a conservative extension of a theory $\mathcal{T}$. The bit definition describes the individual bits of the output, but the theory must also be able to prove the existence of a string collecting all these bits.

However, if $\mathcal{T}$ proves the Φ-**COMP** axiom scheme, then comprehension can be used to construct the output string. Consequently, a polynomially bounded string function whose bit graph is Φ-representable is $\Sigma^B_0(\Phi)$-definable in $\mathcal{T}$. More generally, we have the following lemma.

Lemma 5.20 (Extension by Bit Definition)**.** *Let $\mathcal{T}$ be a theory over $\mathcal{L}$ that contains V^0, and let Φ be a set of $\mathcal{L}$-formulas such that $\Phi \supseteq \Sigma^B_0(\mathcal{L})$. Suppose that $\mathcal{T}$ proves the Φ-**COMP** axiom scheme. Then:*

1. *every polynomially bounded number function whose graph is Φ-representable is $\Sigma^B_0(\Phi)$-definable in $\mathcal{T}$; and*
2. *every polynomially bounded string function whose bit graph is Φ-representable is $\Sigma^B_0(\Phi)$-definable in $\mathcal{T}$.*

In particular, taking $\mathcal{T} = \mathsf{V}^0$ and $\Phi = \Sigma^B_0(\mathcal{L}^2_A)$, the lemma has the following consequence (thanks to $\Sigma^B_0(\Phi) = \Sigma^B_0(\mathcal{L}^2_A)$).

Corollary 5.21. *Every function in FAC^0 is Σ^B_0-definable in V^0.*

We now specialise the preceding notion of bit-definability to formulas in a given language. If a p-bounded string function F is bit-definable by a $\Sigma^B_0(\mathcal{L})$ formula, we say that F is Σ^B_0-*definable from* $\mathcal{L}$. Similarly, a p-bounded number function f is Σ^B_0-*definable from* $\mathcal{L}$ if its graph is represented by a $\Sigma^B_0(\mathcal{L})$ formula.

Corollary 5.22. *Let $\mathcal{T}$ be a theory over $\mathcal{L}$ that contains V^0, and suppose that $\mathcal{T}$ proves the $\Sigma^B_0(\mathcal{L})$-**COMP** axiom scheme. Then a function which is Σ^B_0-definable from $\mathcal{L}$ is $\Sigma^B_0(\mathcal{L})$-definable in $\mathcal{T}$.*

Corollary 5.23. *Suppose that the theory $\mathcal{T}$ proves $\Sigma^B_0(\mathcal{L})$-**COMP** where $\mathcal{L}$ is the vocabulary of $\mathcal{T}$. Then the theory resulting from $\mathcal{T}$ by adding the $\Sigma^B_0(\mathcal{L})$-defining axioms or the $\Sigma^B_0(\mathcal{L})$-bit-defining axioms for a collection of number functions and string functions is a conservative extension of $\mathcal{T}$.*

The following theorem shows that extending a theory with the $\Sigma^B_0(\mathcal{L})$-defining axioms or the $\Sigma^B_0(\mathcal{L})$-bit-defining axioms is not only a conservative extension over V^0, it also proves the $\Sigma^B_0(\mathcal{L})$-**COMP** and $\Sigma^B_0(\mathcal{L})$ induction axioms schemes,

Lemma 5.24 (Σ^B_0-Transformation)**.** *Let $\mathcal{T}$ be a polynomially bounded theory which extends V^0, and assume that the vocabulary $\mathcal{L}$ of $\mathcal{T}$ has the same predicate symbols as $\mathcal{L}^2_A$. Suppose that for every number function f in $\mathcal{L}$, $\mathcal{T}$ proves a $\Sigma^B_0(\mathcal{L}^2_A)$ defining axiom for f, and for every string function F in $\mathcal{L}$, $\mathcal{T}$ proves a $\Sigma^B_0(\mathcal{L}^2_A)$ bit-defining axiom for F. Then for every $i \geq 0$ and every $\Sigma^B_i(\mathcal{L})$ formula φ^+ there is a $\Sigma^B_i(\mathcal{L}^2_A)$ formula φ such that*

$$\mathcal{T} \vdash \varphi^+ \leftrightarrow \varphi.$$

Definition 5.25 ($\Delta^1_1(\mathcal{L})$ and $\Delta^B_1(\mathcal{L})$ Definable Relation)**.** Let $\mathcal{T}$ be a theory over the vocabulary $\mathcal{L}$ and R a relation symbol not in $\mathcal{L}$. We say that R is $\Delta^1_1(\mathcal{L})$-definable (or simply Δ^1_1-definable) in $\mathcal{T}$ if there are $\Sigma^1_1(\mathcal{L})$ formulas $\varphi(\vec{x}, \vec{X})$ and $\psi(\vec{x}, \vec{X})$ such that

$$R(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, \vec{X})$$

holds and

$$\mathcal{T} \vdash \varphi(\vec{x}, \vec{X}) \leftrightarrow \neg\psi(\vec{x}, \vec{X}).$$

We say that R is $\Delta^B_1(\mathcal{L})$-definable (or simply Δ^B_1-definable) in $\mathcal{T}$ if the formulas φ and ψ above are Σ^B_1 formulas.

So for now, what we can conclude is that the number functions in FAC^0 are Σ^B_0-definable in V^0, and string functions in FAC^0 are not only Σ^B_0-definable, but also Σ^B_0-bit-definable in V^0. As a consequence of the witnessing theorem for V^0 (see Theorem V.5.1 in [CN10]), one has the following definability theorem.

Theorem 5.26 (Σ^1_1-Definability Theorem for V^0)**.** *A function is in FAC^0 iff it is Σ^1_1-definable in V^0 iff it is Σ^B_1-definable in V^0 iff it is Σ^B_0-definable in V^0.*

Corollary 5.27. *A relation is in AC^0 iff it is Δ^1_1 definable in V^0 iff it is Δ^B_1 definable in V^0.*

Intuitively, the universal theory $\overline{V}^0$ is a conservative extension over the language $\mathcal{L}_{FAC}$ which contains a function symbol for every FAC^0 function. Also, all those function symbols can be applied in the comprehension axioms scheme, induction scheme, etc.

Lemma 5.28. *$\overline{V}^0$ proves the $\Sigma^B_0(\mathcal{L}_{FAC^0})$-**COMP**, $\Sigma^B_0(\mathcal{L}_{FAC^0})$-**IND**, and $\Sigma^B_0(\mathcal{L}_{FAC^0})$-**MIN** axiom schemes.*

5.1.2 The Theory VNC^2

A layered monotone Boolean circuit C with $d+1$ layers is encoded as follows:

- I (for "inputs") is a string variable, where $|I| \leq n$, that defines the (Boolean) input gates to the circuit.
- G (for "gates") is a string variable such that $G(x, y)$ holds for $x \in [d]$ if the yth gate in layer x is $\wedge$, and is $\vee$ otherwise.

- The wires of C are encoded by a three-dimensional array, namely a string variable E (for "edges") such that $E(z, x, y)$ holds if and only if the output of gate x on layer z is connected to the input of gate y on layer $z + 1$.

To evaluate the gates of C on input I, we compute the values of the gates in each layer, starting from the input layer, in $d + 1$ stages, using the values of the previous layer. The formula $\delta_{LMCV}(n, d, E, G, I, Y)$ below formalises this evaluation procedure (where LMCV stands for "layered monotone circuit value"). The two-dimensional array Y stores the computation results, namely the evaluation string: for $0 \leq z \leq d$, row $Y^{[z]}$ contains the gates on layer z that output 1.

$$\begin{aligned}\delta_{LMCV}(n, d, E, G, I, Y) \equiv \forall x < n \forall z < d((Y(0, x) \leftrightarrow I(x)) \\ \wedge (Y(z + 1, x) \leftrightarrow ((G(z + 1, x) \wedge \forall u < n, E(z, u, x) \rightarrow Y(z, u)) \\ \vee (\neg G(z + 1, x) \wedge \exists u < n, E(z, u, x) \wedge Y(z, u)))))\end{aligned}$$

The following formula states that the circuit with underlying graph (n, d, E) has fan-in at most two:

$$\begin{aligned}\mathit{Fanin2}(n, d, E) \equiv \forall z < d \forall x < n \exists u_1 < n \exists u_2 < n \forall v < n \\ (E(z, v, x) \rightarrow (v = u_1 \vee v = u_2)).\end{aligned}$$

Definition 5.29 (VNC^2)**.** The theory VNC^2 has vocabulary $\mathcal{L}_A^2$ and is axiomatized by the axioms of V^0 and the axiom:

$$\mathit{Fanin2}(n, |n|^2, E) \rightarrow \exists Y \leq \langle |n|^2 + 1, n \rangle \delta_{LMCV}(n, |n|^2, E, G, I, Y).$$

Theorem 5.30. *1. A function is Σ_1^B-definable in VNC^2 if and only if it is in FNC^2.*

2. A relation is Δ_1^1-definable in VNC^2 if and only if it is Δ_1^B-definable in VNC^2 if and only if it is in NC^2.

Definition 5.31 (The universal extension $\overline{\mathsf{VNC}^2}$)**.** Let $\mathcal{L}_{\mathsf{FNC}^2}$ denote the universal language associated with the complexity class NC^2. It extends $\mathcal{L}_A^2$ by string-function symbols representing all string functions in FNC^2; moreover, every number function in FNC^2 is represented by a term of $\mathcal{L}_{\mathsf{FNC}^2}$.

Applying the general construction of [CN10, Chapter IX] to the theory VNC^2, we obtain a universally axiomatized theory $\overline{\mathsf{VNC}^2}$ over the language $\mathcal{L}_{\mathsf{FNC}^2}$. The theory $\overline{\mathsf{VNC}^2}$ contains defining axioms for the new function symbols given by the Cook–Nguyen construction and is a conservative extension of VNC^2.

Theorem 5.32. *The theory $\overline{\mathsf{VNC}^2}$ has the following properties.*

1. *$\overline{\mathsf{VNC}^2}$ is conservative over VNC^2. In particular, every sentence in the original language $\mathcal{L}_A^2$ that is provable in $\overline{\mathsf{VNC}^2}$ is already provable in VNC^2.*

2. *The string functions represented by function symbols of $\mathcal{L}_{\mathsf{FNC}^2}$ are precisely the string functions in FNC^2, and the number functions represented by terms of $\mathcal{L}_{\mathsf{FNC}^2}$ are precisely the number functions in FNC^2.*

3. *$\overline{\mathsf{VNC}^2}$ proves the*

$$\Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})\text{-}\mathbf{COMP}, \qquad \Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})\text{-}\mathbf{IND}, \qquad \Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})\text{-}\mathbf{MIN}$$

axiom schemes.

Theorem 5.33. • *A string function is in* FNC^2 *iff it is represented by a string function symbol in* $\mathcal{L}_{\mathsf{FNC}^2}$.

• *A relation is in* NC^2 *iff it is represented by an open formula of* $\mathcal{L}_{\mathsf{FNC}^2}$ *iff it is represented by a* $\Sigma^B_0(\mathcal{L}_{\mathsf{FNC}^2})$ *formula.*

Theorem 5.34. • *A function is in* FNC^2 *iff it is* $\Sigma^B_1(\mathcal{L}_{\mathsf{FNC}^2})$*-definable in* $\overline{\mathsf{VNC}^2}$ *iff it is* $\Sigma^B_1(\mathcal{L}^2_A)$*-definable in* $\overline{\mathsf{VNC}^2}$.

• *A relation is in* NC^2 *iff it is* $\Delta^B_1(\mathcal{L}_{\mathsf{FNC}^2})$*-definable in* $\overline{\mathsf{VNC}^2}$ *iff it is* $\Delta^B_1(\mathcal{L}^2_A)$*-definable in* $\overline{\mathsf{VNC}^2}$.

5.1.3 Some Basic Formalisation in V^0

In this section, we fix some basic notation for primitive objects coded in V^0. Most formalisations here are routine and can be found in [CN10] (cf. [MT14]). Recall that when speaking about *numbers* in the theory, we mean natural numbers (that is, first-sort objects), and when discussing *strings* we refer to the second-sort objects.

Natural number sequences of constant length. For two numbers x, y let $\langle x, y\rangle := (x+y)(x+y+1)+2y$ be the *pairing function*. We also Σ^B_0-define inductively $\langle v_1, \ldots, v_k\rangle := \langle\langle v_1, \ldots, v_{k-1}\rangle, v_k\rangle$, for every $k > 2$. Then V^0 proves the injectivity of the pairing function and enables us to handle such pairs in a standard way.

Sequences of numbers of variable length. In our two-sorted setting, we represent sequences of natural numbers whose *length is non-constant* using string variables. Using the pairing function we can encode sequences as sets of numbers: a sequence is encoded as a string Z such that the xth number in the sequence is y if the number $\langle x, y\rangle$ is in Z. Formally, we have the following Σ^B_0-defining formula for the number function $\mathbf{seq}(x, Z)$ returning the xth element in the sequence Z:

$$\begin{aligned} y = \mathbf{seq}(x, Z) \leftrightarrow\ & (y < |Z| \wedge Z(\langle x, y\rangle) \wedge \forall z < y \neg Z(\langle x, z\rangle)) \\ & \vee (\forall z < |Z| \neg Z(\langle x, z\rangle) \wedge y = |Z|). \end{aligned} \tag{12}$$

Formula (12) states that the xth element in the sequence coded by Z is y iff $\langle x, y\rangle$ is in Z and no other number smaller than y also "occupies the xth position in the sequence", and that if no number occupies position x then the function returns the length of the string variable Z.

Array of strings. We wish to encode a *sequence* of binary strings as an array, that is, as a two-dimensional array. We use the function $\mathbf{Row}(x, Z)$ to denote the xth string in Z as follows (we follow the treatment in [CN10, Definition V.4.26, p. 114]). The string function $\mathbf{Row}(x, Z)$, abbreviated $Z^{[x]}$, is Σ^B_0-definable in V^0 (and can be added to the theory V^0 to get a conservative extension and used in the induction axiom) using the following bit-definition:

$$\mathbf{Row}(x, Z)(i) \leftrightarrow (i < |Z| \wedge Z(\langle x, i\rangle)).$$

The theory V^0 is a conservative extension of the theory $\mathsf{I}\Delta^0$ [CN10, Thm, V.1.9] (the single-sorted theory of arithmetic axiomatized by induction for Δ_0 formulas over a base theory such as the axioms $\mathbf{B1}, \ldots, \mathbf{B8}$). In particular, recall that V^0 proves the Σ^B_0-**IND** axioms, which include the $I\Delta_0$-**IND** axioms. We can therefore use in V^0 some well-known definitions developed originally in $I\Delta_0$, such as the following.

Graph of 2^n. There is a Δ^B_0-definition of the graph of 2^n such that V^0 proves that it is a partial function that satisfies the defining recurrence $2^0 = 1$, $2^{n+1} = 2 \cdot 2^n$. Also, V^0 proves that if $y = 2^x$, then for any $n < x$, there exists m such that $m = 2^n$. Moreover, if $y = 2^x$ and $z = 2^h$, then $y \cdot z = 2^{x+h}$ (see, e.g., [CN10, III.3, p. 61]). Thus, there is also a well-behaved Δ^B_0-definition of the function $\mathbf{bit}(x, i) = \lfloor x 2^{-i} \rfloor \mod 2$, and $|x| = \min\{n : x < 2^n\}$.

5.2 Some basic formalisation of algebra in VNC^2

In this section we fix some basic notation for algebraic objects coded in VNC^2. Throughout the remainder of the paper, we use a first-sort variable N as the main size parameter; in our applications, N denotes the number of coordinates in the coefficient vector under consideration. We only consider objects whose encodings have size polynomially bounded in N: all first-sort parameters, such as indices, degrees, matrix dimensions, and circuit sizes, are bounded by a fixed polynomial in N, and all second-sort objects are represented by strings of length bounded by a fixed polynomial in N. Large integers and rational numbers are represented in binary, so polynomial boundedness refers to the length of their binary encodings rather than to their numerical values. Under these bounds, the instances of binary integer and rational arithmetic used in the proof, including addition, multiplication, iterated addition and multiplication, division, factorials, and binomial coefficients, are computable by functions in FNC^2 and hence are available in the universal conservative extension $\overline{\mathsf{VNC}^2}$. We use $|\cdot|_1$ for the ordinary absolute value and for its formal counterpart on encoded rational numbers. Rational numbers are encoded as tuples of binary integers. See [Jeř22, Jeř23] for the relevant bounded-arithmetic formalisation of binary arithmetic.

Throughout each formalised algebraic statement, the underlying coefficient field $\mathbb{F}$ is fixed. We use either $\mathbb{Q}$ or a fixed finite field $\mathbb{F}_q$. Elements of $\mathbb{Q}$ are encoded as above, while elements of a fixed finite field $\mathbb{F}_q$ have constant-length encodings. In either case, field addition, multiplication, and equality are computable in FNC^2. Under the polynomial bounds imposed above, all algebraic objects appearing below have encodings of length polynomially bounded in N.

Matrices. We encode an $n \times m$ matrix as an array of n strings, each of which encodes one row of the matrix, and each row is itself an array of m field elements. We only consider matrices whose dimensions and whose total encoding lengths are polynomially bounded in N. The basic matrix operations that we use, including matrix addition, matrix multiplication, and the Kronecker product of a constant number of matrices, are computable by functions in FNC^2. Hence, by the definability theorem for VNC^2, these operations are Σ^B_1-definable in VNC^2 and may be represented by function symbols in the universal conservative extension $\overline{\mathsf{VNC}^2}$. Consequently, when working in $\overline{\mathsf{VNC}^2}$, we may use these matrix operations inside $\Sigma^B_0(\mathcal{L}_{\mathsf{FNC}^2})$ formulas and apply the corresponding comprehension and induction schemes.

Polynomial. Fix a canonical ordering of monomials. For a polynomial f, we write $\mathrm{coeff}_N(f)$ for the vector consisting of the coefficients of the first N monomials in this ordering. Whenever we use this notation, N is chosen large enough that every monomial which may occur in f is among these N monomials, so that $\mathrm{coeff}_N(f)$ determines f. We encode the coefficient vector as a string; throughout the proof, the bit-length of every coefficient and hence the total length of this encoding are polynomially bounded in N.

We say that a polynomial family f is Φ-definable if its N-coordinate coefficient-vector function is Φ-bit-definable with output length polynomially bounded in N. In particular, the Permanent and Iterated Matrix Multiplication polynomials have Σ^B_0-definable coefficient-vector functions. For the Permanent polynomial

$$\mathrm{perm}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)},$$

the coefficient of a monomial is non-zero if and only if the monomial contains exactly one variable from every row and every column. For

$$\mathrm{IMM}_{n,d}(X) = \left(X^{(1)} X^{(2)} \cdots X^{(d)}\right)_{1,1},$$

the coefficient of a monomial is non-zero if and only if the monomial contains exactly one variable from each matrix $X^{(t)}$, the selected variable in $X^{(1)}$ has row index 1, the selected variable in $X^{(d)}$

has column index 1, and, for every $t < d$, the column index selected in $X^{(t)}$ equals the row index selected in $X^{(t+1)}$. These conditions are decidable in AC^0 and therefore in NC^2. We use

$$\mathbf{Perm}(n, N) \coloneqq \mathrm{coeff}_N(\mathrm{perm}_n) \qquad \text{and} \qquad \mathbf{IMM}(n, d, N) \coloneqq \mathrm{coeff}_N(\mathrm{IMM}_{n,d})$$

for the corresponding string functions. Whenever these functions are used, the parameters n, d and the output encoding lengths are polynomially bounded in N.

Algebraic circuit. Algebraic circuits are encoded using strings in the theory as follows [TC21]:

1. a sequence V of node labels $v_1, v_2, \ldots, v_k$, where each label is a natural number;
2. a string of gates G, where each gate is encoded by a pair (v, t) of natural numbers: v identifies a node in V, while t specifies whether the node is a $+$-gate, a $\times$-gate, or an input gate;
3. a two-dimensional string I encoding the input gates, where an input gate encodes either a variable x_j or an element of the fixed coefficient field $\mathbb{F}$; and
4. a string E encoding directed edges between nodes, where $(u, v) \in E$ means that there is an edge from u to v.

Throughout the formalisation, the lengths of V, G, I, E, the number of circuit nodes, the indices of the variables occurring in the circuit, and the bit-lengths of all scalar constants are polynomially bounded in N. Thus a circuit of the type considered in the proof is represented by a tuple of strings of length polynomial in N.

Rank function. We use the standard NC^2 algorithms for matrix rank. For rational matrices, Ken and Kuroda [KK23], building on Tzameret and Cook [TC21], formalised the relevant rank algorithm in VNC^2 and proved the usual linear-algebraic characterisations of rank. We write $\mathbf{rank}_{\mathbb{F}}(M)$ for the rank over the fixed underlying field. In particular, the rank is bounded by the number of rows and by the number of columns, and it satisfies the standard rank identities and inequalities used below. More precisely, Ken and Kuroda showed that the rank $\mathbf{rank}(M)$ of a matrix M with $\mathbb{F}$-coefficients coincides with [KK23, Lemmas 60, 62 and Theorem 65]:

- the number of linearly independent row vectors;
- the number of linearly independent column vectors;
- the largest order of a non-zero minor.

5.3 Formalisation of [LST25] and [For24] in VNC^2

In this section, the notations and concepts are taken from [LST25] and [For24].
We present a proof outline here:

1. We define a rank measure, the relative rank, in the theory.
2. By a basic rank argument, the coefficient matrix of the word polynomial P_w induced by a b-unbiased word w has high relative rank.
3. We then show that, whenever the full coefficient vector has polynomially bounded encoding length in the parameter regime considered below, polynomial identities involving the relevant constant-depth circuits can be proved in VNC^2 by reasoning about their full coefficient vectors.

4. With the ability to prove polynomial identities of constant-depth circuits, we show that for any constant-depth circuit $C(\overline{X})$ that computes a polynomial $f(\overline{X})$, the set-multilinear projection $\pi_{sm}(f)$ can be computed by a constant-depth set-multilinear circuit C' of size $\mathrm{poly}(s, d^d)$ over the sets of variables $(X_1, \ldots, X_d)$. In particular, for a set-multilinear polynomial f, a sufficiently strong lower bound against constant-depth set-multilinear circuits implies a lower bound against general constant-depth circuits computing f.

5. We prove an upper bound on the relative rank of any constant-depth set-multilinear circuit C' in terms of the size of C'. Roughly speaking, the relative rank of small-size constant-depth set-multilinear circuits cannot be too big.

6. Combining the above lower bound for the relative rank of P_w with the upper bound for the relative rank of small-size constant-depth set-multilinear circuits, we obtain a size lower bound against constant-depth set-multilinear circuits that compute P_w. Moreover, this lower bound is sufficient to imply a superpolynomial-size lower bound against constant-depth circuits.

7. Finally, we transfer the lower bound for P_w to the Permanent polynomial using the explicit polynomial projection constructed later in the proof from a suitable Permanent polynomial to P_w.

Throughout the formalisation, we use a first-sort variable N as the main size parameter. The parameter N denotes the number of coordinates in the coefficient vector under consideration. For polynomials in n variables of total degree at most d, this means $N \geq \binom{n+d}{d}$.

All auxiliary first-sort quantities occurring in the formalisation are bounded by a fixed polynomial in N, and every second-sort object is encoded by a string of length bounded by a fixed polynomial in N. This convention guarantees that all objects manipulated in the formal proof have polynomial-size encodings with respect to N. It does not change the size parameter in the algebraic circuit lower bound itself.

We retain the parameter choices from the algebraic lower-bound argument. In particular,

$$n(N) = \lfloor \log_2 N \rfloor, \quad d(N) = \lfloor c \log_2 n(N) \rfloor, \quad k(N) = 10d(N).$$

where c is a fixed rational constant. These relations can be represented by relations in NC^2 and all the resulting first-sort parameters are polynomially bounded in N.

We will study word polynomials indexed by words over a finite alphabet depending on the parameter k. Let $a(k) = \left\lfloor \frac{k}{\sqrt{2}} \right\rfloor$ and define the two-element alphabet $A(k) := \{a(k), -k\}$. We then consider words $w = (w_1, \ldots, w_d) \in A(k)^d$ and the associated word polynomials P_w, which will be defined below. The integer $a(k)$ can be defined without introducing the irrational number $1/\sqrt{2}$ into the language: it is the unique nonnegative integer a satisfying $2a^2 \leq k^2 < 2(a+1)^2$. Thus every entry of w has absolute value at most k. The word polynomial P_w is defined for every $w \in A(k)^d$; when proving the lower bound, we will additionally require w to be k-unbiased.

We will use Δ to denote the product-depth of algebraic circuits, which is constant. We will consider algebraic circuits of size polynomially bounded in the number variables n. Namely, if s is a fixed standard polynomial, we consider circuits of size at most $s(n)$. The strings V, G, I, E encoding such a circuit therefore have length at most $\mathrm{poly}(s(n))$.

5.3.1 Exponentiation in VTC^0

The proof of [LST25] applies one elementary analytic function, the exponential function, to rational inputs, producing values that need not be rational. Jeřábek [Jeř23] formalises the required elementary analytic functions in VTC^0 and gives uniform TC^0 rational approximations of these functions. We give a short exposition of the results necessary for our work. We restrict attention to real inputs, although the results in [Jeř23] are proved more generally for complex inputs.

Let $D \subseteq \mathbb{R}$ and let $f : D \to \mathbb{R}$. An *additive* TC^0 *approximation* of f is a TC^0 function

$$f_+ : \mathbb{Q} \times \mathbb{N} \to \mathbb{Q}$$

such that

$$|f_+(z,t) - f(z)|_1 \leq 2^{-t}$$

for every $z \in D \cap \mathbb{Q}$ and every first-sort precision parameter t. A *multiplicative* TC^0 *approximation* of f is a TC^0 function

$$f_\times : \mathbb{Q} \times \mathbb{N} \to \mathbb{Q}$$

such that

$$|f_\times(z,t) - f(z)|_1 \leq 2^{-t}|f(z)|_1$$

for every $z \in D \cap \mathbb{Q}$ and every first-sort precision parameter t. Equivalently, whenever $f(z) \neq 0$,

$$\left|\frac{f_\times(z,t)}{f(z)} - 1\right|_1 \leq 2^{-t}.$$

The approximation algorithms for the exponential function require an additional magnitude parameter, since the binary length of the output may grow with the magnitude of the input. Let $P(z,r)$ be a property of a rational input z and a first-sort parameter r. We say that a TC^0 function $f_+(z,r,t)$ is an *additive approximation of* $f(z)$ *parametrised by* r *subject to* $P(z,r)$ if

$$P(z,r) \implies |f_+(z,r,t) - f(z)|_1 \leq 2^{-t}$$

for every $z \in D \cap \mathbb{Q}$ and all first-sort parameters r, t. We additionally require that for every $z \in D \cap \mathbb{Q}$, there exists a first-sort parameter r such that $P(z,r)$. Multiplicative approximation parametrised by r is defined analogously.

In the applications below, the binary encoding length of z and the numerical values of the first-sort parameters r, t are all bounded by fixed polynomials in the global size parameter N. Therefore every TC^0 approximation function used below has output encoding length polynomially bounded in N.

Definition 5.35 (The exponential function)**.** For $z \in \mathbb{Q}$ and a first-sort number m, define the truncated exponential series

$$e(z,m) \coloneqq \sum_{j<m} \frac{z^j}{j!}.$$

For semantic purposes, we use $\exp : \mathbb{R} \to \mathbb{R}_{>0}$ for the ordinary real exponential function. For every rational z,

$$\exp z = \lim_{m\to\infty} e(z,m).$$

The exact real number $\exp z$ is used only in the semantic correctness statements for the rational approximation functions below; the bounded-arithmetic theory itself manipulates their rational outputs. We use the standard identity

$$\exp(z+w) = \exp z \, \exp w$$

for all $z, w \in \mathbb{R}$.

Lemma 5.36 (Existence of TC^0 approximations of exp [Jeř23])**.** *There are* TC^0 *functions* $E_\times(z,r,t)$ *and* $E_+(z,r,t)$ *such that, for all* $z \in \mathbb{Q}$ *and all first-sort parameters* r, t*:*

1. *if* $|z|_1 \leq r$*, then*

$$|E_\times(z,r,t) - \exp z|_1 \leq 2^{-t} \exp z;$$

hence $E_\times(z,r,t)$ *is a multiplicative approximation of* $\exp z$ *parametrised by* r *subject to* $|z|_1 \leq r$*;*

2. if $z \le r$, then

$$|E_+(z,r,t) - \exp z|_1 \le 2^{-t};$$

hence $E_+(z,r,t)$ is an additive approximation of $\exp z$ parametrised by r subject to $z \le r$.

In the proof, we use a rational approximation of 2^z rather than directly approximating e^z. Since $2^z = \exp(z \ln 2)$, we also require a rational approximation of $\ln 2$. The construction of the logarithm function in [Jeř23] is more involved, so we state only the facts that we need.

Lemma 5.37 ([Jeř23])**.** *The real logarithm function $\ln : \mathbb{R}_{>0} \to \mathbb{R}$ is the inverse of the real exponential function. Thus $\exp(\ln z) = z$ for every $z > 0$, and $\ln(\exp z) = z$ for every $z \in \mathbb{R}$.*

Moreover, $\ln z$ admits uniform TC^0 additive and multiplicative rational approximations on positive rational inputs. In particular, there are TC^0 functions $L_+(z,t)$ and $L_\times(z,t)$ such that for every $z \in \mathbb{Q}_{>0}$ and every first-sort precision parameter t,

$$|L_+(z,t) - \ln z|_1 \le 2^{-t}$$

and

$$|L_\times(z,t) - \ln z|_1 \le 2^{-t} |\ln z|_1.$$

Remark 5.38. In [Jeř23], the functions exp and log are developed more generally over complex inputs. We need only their restriction to real inputs and, for the logarithm, only the value $\ln 2$. The exact real-valued functions are used only to state the semantic guarantees of the approximation procedures, while the theory itself works with their rational-valued TC^0 approximations.

We now construct the approximation of the powering function

$$\mathrm{pow}(z) \coloneqq 2^z = \exp(z \ln 2).$$

Lemma 5.39. *Define*

$$P_\times(z,r,t) \coloneqq E_\times(zL_\times(2, t+r+4), 2r, t+2)\,.$$

Then $P_\times$ is a TC^0 multiplicative approximation of $\mathrm{pow}(z) = 2^z$, parametrised by r subject to $|z|_1 \le r$. More precisely,

$$|P_\times(z,r,t) - 2^z|_1 \le 2^{-t} 2^z$$

whenever $z \in \mathbb{Q}$ and $|z|_1 \le r$.

Proof. Let $s = t + r + 4$. By the multiplicative approximation property of $L_\times$, there exists a real number δ_1 such that $L_\times(2,s) = (\ln 2)(1+\delta_1)$, with $|\delta_1|_1 \le 2^{-s}$. Since $s \ge 4$ and $\ln 2 < 1$, we have $|L_\times(2,s)|_1 \le (\ln 2)(1 + 2^{-4}) < 2$. Consequently, if $|z|_1 \le r$, then $|zL_\times(2,s)|_1 \le 2r$, so the magnitude parameter $2r$ is sufficient for the call to $E_\times$.

By the multiplicative approximation property of $E_\times$, there exists a real number δ_2 such that

$$P_\times(z,r,t) = \exp\big(zL_\times(2,s)\big)(1+\delta_2), \qquad |\delta_2|_1 \le 2^{-(t+2)}.$$

Substituting the expression for $L_\times(2,s)$ gives

$$zL_\times(2,s) = z\ln 2 + z \ln 2\, \delta_1.$$

Hence, using $\exp(a+b) = \exp a \exp b$,

$$P_\times(z,r,t) = 2^z \exp(z \ln 2\, \delta_1)(1+\delta_2).$$

Therefore

$$\frac{P_\times(z,r,t)}{2^z} = (1+\delta_2)e^\tau,$$

where $\tau := z \ln 2\, \delta_1$. Since $|z|_1 \le r$, $\ln 2 < 1$, and $|\delta_1|_1 \le 2^{-(t+r+4)}$, we obtain

$$|\tau|_1 \le r\, 2^{-(t+r+4)}.$$

Using $r \le 2^r$ for every natural number r, we get

$$|\tau|_1 \le 2^{-(t+4)}.$$

In particular $|\tau|_1 \le 1$. For every real x with $|x|_1 \le 1$,

$$|e^x - 1|_1 \le 2|x|_1,$$

and therefore

$$|e^\tau - 1|_1 \le 2^{-(t+3)}.$$

We now estimate the relative error:

$$\begin{aligned}\frac{P_\times(z,r,t)}{2^z} - 1 &= (1+\delta_2)e^\tau - 1 \\ &= \delta_2 + (e^\tau - 1) + \delta_2(e^\tau - 1).\end{aligned}$$

Hence,

$$\begin{aligned}\left|\frac{P_\times(z,r,t)}{2^z} - 1\right|_1 &\le |\delta_2|_1 + |e^\tau - 1|_1 + |\delta_2|_1 |e^\tau - 1|_1 \\ &\le 2^{-(t+2)} + 2^{-(t+3)} + 2^{-(t+2)}2^{-(t+3)} \\ &\le 2^{-t}.\end{aligned}$$

Multiplying by $2^z > 0$ yields

$$|P_\times(z,r,t) - 2^z|_1 \le 2^{-t}2^z.$$

Thus $P_\times$ is a multiplicative TC^0 approximation of pow. □

Since $P_\times$ is obtained by composition of TC^0 functions and rational arithmetic, it is itself a TC^0 function. Hence $P_\times \in \mathsf{FTC}^0 \subseteq \mathsf{FNC}^2$ and is available in the universal conservative extension $\overline{\mathsf{VNC}^2}$. In all applications below, the encoding length of z and the first-sort parameters r, t are polynomially bounded in the global size parameter N. Therefore the binary encoding of $P_\times(z,r,t)$ is also polynomially bounded in N.

5.3.2 Word polynomials

Consider the set of words on an alphabet $A \subseteq \mathbb{Z} \setminus 0$. Let $w = (w_1, \ldots, w_d) \in A^d$. For a subset $S \subseteq [d]$, let $w_S := \sum_{i \in S} w_i$. We define $\mathcal{P}_w = \{i \in [d] : w_i \ge 0\}$ and $\mathcal{N}_w = \{i \in [d] : w_i < 0\}$, the positive and negative index sets of w, respectively. We say that $w \in A^d$ is *b-unbiased* if $|w_{[t]}|_1 \le b$ for every $t \le d$. Note that $|\cdot|_1$ denotes the ordinary absolute value.

Given w, we denote by $\overline{X}(w) = (X(w_1), \ldots, X(w_d))$ a tuple of d sets of disjoint variables, where $\#X(w_i) = 2^{|w_i|_1}$. We denote by $\mathbb{F}_{sm}[T]$ the space of set-multilinear polynomials over a tuple T of sets of variables. Hence $\mathbb{F}_{sm}[\overline{X}(w)]$ consists of polynomials in which every monomial contains exactly one variable from each set $X(w_i)$.

Since $X(w_i)$ contains $2^{|w_i|_1}$ variables, we label its variables by strings in $\{0,1\}^{|w_i|_1}$. Let $M_w^{\mathcal{P}}$ and $M_w^{\mathcal{N}}$ denote the sets of set-multilinear monomials over the positive and negative variable sets, respectively. Given a monomial $m \in \mathbb{F}_{sm}[\overline{X}(w)]$, let m_+ denote its positive part in $M_w^{\mathcal{P}}$ and let m_- denote its negative part in $M_w^{\mathcal{N}}$. We use $\sigma(m_+)$ to denote the Boolean string associated with m_+, and $\sigma(m_-)$ to denote the Boolean string associated with m_-.

More precisely, if $j_1 < \cdots < j_t$ and $m' = x_{\sigma_1}^{(j_1)} x_{\sigma_2}^{(j_2)} \cdots x_{\sigma_t}^{(j_t)}$ with $x_{\sigma_i}^{(j_i)} \in X_{j_i}$ and $\sigma_i \in \{0,1\}^{|w_{j_i}|_1}$ for each $i \in [t]$, then $\sigma(m')$ is defined to be $\sigma_1 \cdots \sigma_t$.

The length of $\sigma(m_+)$ is $\sum_{i\in\mathcal{P}_w} w_i$, whereas the length of $\sigma(m_-)$ is $-\sum_{i\in\mathcal{N}_w} w_i$.

Consequently, the difference between these two lengths is $\left|\sum_{i\in[d]} w_i\right|_1 = |w_{[d]}|_1$. In particular, if w is b-unbiased, this difference is at most b.

For binary strings x and y, write $x \sim y$ if the shorter of x, y is a prefix of the longer one. The word polynomial P_w is

$$P_w = \sum_{\substack{m\in\mathbb{F}_{sm}[\overline{X}(w)]\\ \sigma(m_+)\sim\sigma(m_-)}} m.$$

where m is a monomial.

5.3.3 Word polynomials are Σ_1^B-definable in VNC^2

Let $L(w) \coloneqq \sum_{j=1}^{d} |w_j|_1$ and $N_{sm}(w) \coloneqq 2^{L(w)}$. Thus $N_{sm}(w)$ is the number of set-multilinear monomials over $\overline{X}(w)$. We choose the parameters n, d, k as functions of the global parameter N so that $L(w) \le kd \le \lfloor \log_2 N \rfloor$, and hence $N_{sm}(w) = 2^{L(w)} \le N$. In particular, every number and string occurring in the definition below is polynomially bounded in N.

We order the set-multilinear monomials over $\overline{X}(w)$ by identifying them with the binary strings in $0, 1^{L(w)}$, as in the definition of P_w above. We define the string function $\mathbf{WordP}(w, d, N)$ to be the N-coordinate coefficient vector of P_w in this ordering, padded by zeros after the first $N_{sm}(w)$ coordinates.

For $i < N_{sm}(w)$, let $u = \mathbf{binary}(i, L(w))$ be the binary representation of i of length exactly $L(w)$, padded with leading zeros when necessary. Thus

$$\mathbf{binary}(K, L)(j) = \left\lfloor \frac{K}{2^{L-1-j}} \right\rfloor \bmod 2 \qquad \text{for } j = 0, \ldots, L-1,$$

whenever $K < 2^L$.

We partition u into d consecutive blocks such that $u = u_1 u_2 \cdots u_d$ and $|u_j| = |w_j|_1$.Let

$$\mathcal{P}_w = j_1 < j_2 < \cdots < j_p, \qquad \mathcal{N}_w = k_1 < k_2 < \cdots < k_q,$$

we define

$$u^{\mathcal{P}} \coloneqq u_{j_1} u_{j_2} \cdots u_{j_p}, \qquad u^{\mathcal{N}} \coloneqq u_{k_1} u_{k_2} \cdots u_{k_q}.$$

where the blocks are concatenated in increasing order of their indices. Thus $u^{\mathcal{P}}$ and $u^{\mathcal{N}}$ are precisely the strings $\sigma(m_+)$ and $\sigma(m_-)$ associated with the set-multilinear monomial indexed by i.

We define the prefix relation by

$$\mathbf{Prefix}(x, y) \coloneqq |x| \le |y| \wedge \forall j < |x|\, (x(j) \leftrightarrow y(j)).$$

Then the bit graph of **WordP** is given by

$$\begin{aligned}\mathbf{WordP}(w, d, N)(i) \leftrightarrow\ & i < N \wedge i < N_{sm}(w) \\ & \wedge \Big(\mathbf{Prefix}(\mathbf{binary}(i, L(w))^{\mathcal{P}}, \mathbf{binary}(i, L(w))^{\mathcal{N}}) \\ & \quad \vee \mathbf{Prefix}(\mathbf{binary}(i, L(w))^{\mathcal{N}}, \mathbf{binary}(i, L(w))^{\mathcal{P}})\Big).\end{aligned}$$

For $N_{sm}(w) \le i < N$, the ith bit of $\mathbf{WordP}(w, d, N)$ is therefore 0.

The functions used above—computing $L(w)$, extracting the blocks of $\mathbf{binary}(i, L(w))$, concatenating the positive and negative blocks, and checking the prefix relation—are computable in FNC^2. Moreover, the output length of $\mathbf{WordP}(w, d, N)$ is at most N. Therefore $\mathbf{WordP} \in \mathsf{FNC}^2$.

All the operations appearing in the above definition, including **binary**, block extraction, concatenation, and the prefix test, are computable in FNC^2. Hence they are available as function

symbols in the universal language $\mathcal{L}_{\mathsf{FNC}^2}$. Therefore, the displayed formula gives a $\Sigma^B_0(\mathcal{L}_{\mathsf{FNC}^2})$ bit-definition of the string function $\mathbf{WordP}(w, d, N)$ in $\overline{\mathsf{VNC}^2}$.

Since the output length of $\mathbf{WordP}(w, d, N)$ is bounded by N, the corresponding string exists by $\Sigma^B_0(\mathcal{L}_{\mathsf{FNC}^2})$-comprehension in $\overline{\mathsf{VNC}^2}$. By conservativity and the Σ^B_0-transformation theorem, this definition can be translated back to the language of VNC^2. In particular, $\mathbf{WordP}$ is Σ^B_1-definable in VNC^2.

5.3.4 Relative rank

For $f \in \mathbb{F}_{sm}[\overline{X}(w)]$, define $M_w(f)$ to be the coefficient matrix whose rows are indexed by $M^{\mathcal{P}}_w$, whose columns are indexed by $M^{\mathcal{N}}_w$, and whose (m_1, m_2)-entry is the coefficient of the monomial $m_1 m_2$ in f.

Let $L(w) \coloneqq \sum_{i=1}^{d} |w_i|_1$ and $s(w) \coloneqq w_{[d]} = \sum_{i=1}^{d} w_i$. Then $|M^{\mathcal{P}}_w| = 2^{\sum_{i \in \mathcal{P}_w} w_i}$, and $|M^{\mathcal{N}}_w| = 2^{-\sum_{i \in \mathcal{N}_w} w_i}$, and consequently $|M^{\mathcal{P}}_w| \cdot |M^{\mathcal{N}}_w| = 2^{L(w)}$. Moreover, $\min\{|M^{\mathcal{P}}_w|, |M^{\mathcal{N}}_w|\} = 2^{\frac{L(w)-|s(w)|_1}{2}}$. Notice that $L(w) - |s(w)|_1$ is even, so the exponent in the last expression is an integer.

The relative rank of f is defined as

$$\mathrm{relrk}_w(f) \coloneqq \frac{\mathrm{Rank}(M_w(f))}{\sqrt{|M^{\mathcal{P}}_w| \cdot |M^{\mathcal{N}}_w|}} = \frac{\mathrm{Rank}(M_w(f))}{2^{L(w)/2}} \leq 1.$$

As the denominator need not be rational, so inside VNC^2 we instead use the rational approximation $P_\times$ defined above.

Both dimensions of $M_w(f)$ are at most N, and the matrix has at most N^2 entries. Thus its encoding has length polynomially bounded in N. If Y is the N-coordinate coefficient vector of a set-multilinear polynomial, we also write $M_w(Y)$ for the matrix obtained from Y by extracting the corresponding coefficients. This matrix-extraction operation is in FNC^2.

We define the rational relative-rank function used in VNC^2 by

$$\mathbf{relrank}(w, d, N, M) \coloneqq \frac{\mathbf{rank}(M)}{P_\times\left(\frac{L(w)}{2}, N, N\right)}.$$

Since $L(w) \leq N$, the magnitude parameter N is valid. We also use N as the precision parameter. Put $\epsilon_N \coloneqq 2^{-N}$. By the multiplicative approximation property of $P_\times$,

$$(1 - \epsilon_N)2^{L(w)/2} \leq P_\times\left(\frac{L(w)}{2}, N, N\right) \leq (1 + \epsilon_N)2^{L(w)/2}.$$

For $N \geq 1$, the denominator is therefore positive.

By the rank properties established in [KK23], VNC^2 proves the following rational analogues of the standard properties of relative rank.

Claim 5.40. Let $w \in A^d$ satisfy $N_{sm}(w) \leq N$.

- **Imbalance.** For every $f \in \mathbb{F}_{sm}[\overline{X}(w)]$,

$$\mathbf{relrank}(w, d, N, M_w(f)) \leq \frac{2^{(L(w)-|s(w)|_1)/2}}{P_\times(L(w)/2, N, N)}.$$

  Moreover,

$$\mathbf{relrank}(w, d, N, M_w(f)) \leq \frac{1 + \epsilon_N}{1 - \epsilon_N} \frac{1}{P_\times(|s(w)|_1/2, N, N)}.$$

- **Subadditivity.** For every $f, g \in \mathbb{F}_{sm}[\overline{X}(w)]$,

$$\mathbf{relrank}(w, d, N, M_w(f + g)) \leq \mathbf{relrank}(w, d, N, M_w(f)) + \mathbf{relrank}(w, d, N, M_w(g)).$$

- **Multiplicativity.** Suppose $f = f_1 f_2 \cdots f_t$, where $(S_1, \ldots, S_t)$ is a partition of $[d]$ and $f_i \in \mathbb{F}_{sm}[\overline{X}(w \restriction S_i)]$ for every $i \in [t]$. Then

$$\begin{aligned}\mathbf{relrank}(w, d, N, M_w(f)) &\leq \frac{(1+\epsilon_N)^t}{1-\epsilon_N} \prod_{i=1}^{t} \mathbf{relrank}(w \restriction S_i, |S_i|, N, M_{w\restriction S_i}(f_i)) \\ &\leq \left(\frac{1+\epsilon_N}{1-\epsilon_N}\right)^t \prod_{i=1}^{t} \mathbf{relrank}(w \restriction S_i, |S_i|, N, M_{w\restriction S_i}(f_i)) \,.\end{aligned}$$

Proof. **Imbalance.** Since $\mathbf{rank}(M_w(f)) \leq \min\{|M_w^{\mathcal{P}}|, |M_w^{\mathcal{N}}|\}$, we have $\mathbf{rank}(M_w(f)) \leq 2^{(L(w)-|s(w)|_1)/2}$. Dividing by the positive rational number $P_\times(L(w)/2, N, N)$ gives

$$\mathbf{relrank}(w, d, N, M_w(f)) \leq \frac{2^{(L(w)-|s(w)|_1)/2}}{P_\times(L(w)/2, N, N)}.$$

From the approximation bounds, $P_\times(L(w)/2, N, N) \geq (1-\epsilon_N)2^{L(w)/2}$ and $P_\times(|s(w)|_1/2, N, N) \leq (1+\epsilon_N)2^{|s(w)|_1/2}$. Therefore, we have

$$\begin{aligned}\mathbf{relrank}(w, d, N, M_w(f)) &\leq \frac{2^{-|s(w)|_1/2}}{1-\epsilon_N} \\ &\leq \frac{1+\epsilon_N}{1-\epsilon_N} \frac{1}{P_\times(|s(w)|_1/2, N, N)}.\end{aligned}$$

Subadditivity. The matrices satisfy $M_w(f+g) = M_w(f) + M_w(g)$, and matrix rank is subadditive, $\mathbf{rank}(M_w(f+g)) \leq \mathbf{rank}(M_w(f)) + \mathbf{rank}(M_w(g))$. Since all three relative ranks have the same positive denominator $P_\times(L(w)/2, N, N)$, the desired inequality follows.

Multiplicativity. Up to permutations of rows and columns,

$$M_w(f_1 \cdots f_t) = \bigotimes_{i=1}^{t} M_{w\restriction S_i}(f_i).$$

Hence

$$\mathbf{rank}(M_w(f_1 \cdots f_t)) = \prod_{i=1}^{t} \mathbf{rank}(M_{w\restriction S_i}(f_i)).$$

Set

$$L_i \coloneqq \sum_{j \in S_i} |w_j|_1.$$

Since $\sum_i L_i = L(w)$, we obtain

$$\begin{aligned}&\mathbf{relrank}(w, d, N, M_w(f_1 \cdots f_t)) \\ &= \prod_{i=1}^{t} \mathbf{relrank}(w \restriction S_i, |S_i|, N, M_{w\restriction S_i}(f_i)) \cdot \frac{\prod_{i=1}^{t} P_\times(L_i/2, N, N)}{P_\times(L(w)/2, N, N)}.\end{aligned}$$

The approximation bounds imply

$$\prod_{i=1}^{t} P_\times(L_i/2, N, N) \leq (1+\epsilon_N)^t 2^{L(w)/2}$$

and

$$P_\times(L(w)/2, N, N) \geq (1-\epsilon_N)2^{L(w)/2}.$$

Thus

$$\frac{\prod_{i=1}^{t} P_\times(L_i/2, N, N)}{P_\times(L(w)/2, N, N)} \leq \frac{(1+\epsilon_N)^t}{1-\epsilon_N},$$

which proves the first inequality. The second follows because $(1-\epsilon_N)^{t-1} \leq 1$. □

5.3.5 Word polynomials have high rank

The matrix $M_w(P_w)$ has full rank. Indeed, suppose $|M_w^{\mathcal{P}}| \geq |M_w^{\mathcal{N}}|$. Then $\sigma(m_-)$ is no longer than $\sigma(m_+)$. For every column indexed by $m_- \in M_w^{\mathcal{N}}$, the non-zero entries are precisely the rows whose associated strings extend $\sigma(m_-)$. The supports of two distinct columns are disjoint, so the columns are linearly independent over every field. Hence $\mathrm{Rank}(M_w(P_w)) = |M_w^{\mathcal{N}}|$ The case $|M_w^{\mathcal{P}}| < |M_w^{\mathcal{N}}|$ is symmetric, using the rows instead. Therefore

$$\mathrm{Rank}(M_w(P_w)) = \min\{|M_w^{\mathcal{P}}|, |M_w^{\mathcal{N}}|\} = 2^{(L(w)-|s(w)|_1)/2}.$$

Consequently, for the ordinary relative-rank measure, $\mathrm{relrk}_w(P_w) = 2^{-|s(w)|_1/2}$. In particular, if w is b-unbiased, then $|s(w)|_1 \leq b$ and $\mathrm{relrk}_w(P_w) \geq 2^{-b/2}$.

Inside VNC^2, we use the rational approximation instead. Since

$$\mathbf{rank}(M_w(\mathbf{WordP}(w, d, N))) = 2^{(L(w)-|s(w)|_1)/2},$$

we have

$$\mathbf{relrank}(w, d, N, M_w(\mathbf{WordP}(w, d, N))) = \frac{2^{(L(w)-|s(w)|_1)/2}}{P_\times(L(w)/2, N, N)}.$$

If w is b-unbiased, then

$$\mathbf{relrank}(w, d, N, M_w(\mathbf{WordP}(w, d, N))) \geq \frac{1-\epsilon_N}{1+\epsilon_N} \frac{1}{P_\times(b/2, N, N)}.$$

We define the predicate

$$\mathbf{unbiased}(w, d, b) \coloneqq \forall t \leq d \left| \sum_{j=1}^{t} w_j \right|_1 \leq b,$$

where each w_j is obtained from the corresponding segment of the string w. This is a $\Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})$ predicate.

Finally, fix the constant c occurring in the lower-bound construction. Choose a standard constant N_0 sufficiently large that, for every $N \geq N_0$, if

$$n(N) \coloneqq \lfloor \log_2 N \rfloor, \qquad d(N) \coloneqq \lfloor c \log_2 n(N) \rfloor, \qquad k(N) \coloneqq 10d(N),$$

then

$$k(N)d(N) \leq n(N).$$

For every $w \in A(k(N))^{d(N)}$ we then have $L(w) \leq k(N)d(N) \leq n(N)$, and hence $N_{sm}(w) = 2^{L(w)} \leq 2^{n(N)} \leq N$. Thus all coefficient vectors and coefficient matrices occurring above have length polynomially bounded in N.

With these definitions, VNC^2 proves that the coefficient matrix of the word polynomial has high rank:

$$\begin{aligned} \forall N \geq N_0 \; \forall w \in A(k(N))^{d(N)}, \qquad & \mathbf{unbiased}(w, d(N), k(N)) \longrightarrow \\ & \mathbf{relrank}(w, d(N), N, M_w(\mathbf{WordP}(w, d(N), N))) \\ & \qquad \geq \frac{1-2^{-N}}{1+2^{-N}} \frac{1}{P_\times(k(N)/2, N, N)}. \end{aligned}$$

All quantities in this displayed statement are rational or integer-valued and have encodings of length polynomially bounded in N.

5.3.6 The algorithm computing the coefficient vector of a constant-depth circuit

Before proving the field-independent set-multilinearization theorem of Forbes [For24], we first formalise coefficient extraction in VNC^2.

Throughout this section, N is the global size parameter. We choose $n = n(N) = \lfloor \log_2 N \rfloor$ and $d = d(N) = \lfloor c \log_2 n \rfloor$ where c is a fixed constant. All circuits to which the coefficient-extraction procedure is applied have encoding length polynomially bounded in N. Let C be such a circuit, and let $q(C)$ denote the number of variables occurring in C. The number of monomials in these variables of total degree at most d is $D(C,d) = \binom{q(C)+d}{d}$. For all applications below, the parameters are chosen so that $D(C,d) \le N$. Indeed, the number of variables and the circuit size are bounded by a fixed polynomial in $n = \Theta(\log N)$, while $d = O(\log n)$, and therefore

$$\binom{q(C)+d}{d} \le (q(C)+d)^d = 2^{O((\log n)^2)} \le N$$

for all sufficiently large N. For a polynomial f in the variables of C, let $\mathrm{coeff}_{\le d,N}(f)$ denote its coefficient vector on all monomials of total degree at most d, in some fixed ordering, padded with zeros to N coordinates.

We define an FNC^2 string function

$$\mathbf{Coeff}(V, G, I, E, d, N)$$

which, given an encoding (V, G, I, E) of a constant-depth algebraic circuit C, outputs $\mathrm{coeff}_{\le d,N}(C)$. Thus **Coeff** does not attempt to compute coefficients of monomials of degree greater than d.

We compute the truncated coefficient vector gate by gate. For a gate v, write C_v for the polynomial computed at v, and let $\mathrm{coeff}_{\le d,N}(C_v)$ denote its truncated coefficient vector.

- If v is an input gate representing a scalar a, then $\mathrm{coeff}_1(C_v) = a$, and every other coefficient is zero.
- If v is an input gate representing a variable x_j, then $\mathrm{coeff}_{x_j}(C_v) = 1$, and every other coefficient is zero.
- If v is a $+$-gate with incoming gates $u_1, \dots, u_t$, then for every monomial M of degree at most d,
$$\mathrm{coeff}_M(C_v) = \sum_{i=1}^{t} \mathrm{coeff}_M(C_{u_i}).$$
- Suppose that v is a $\times$-gate with incoming gates $u_1, \dots, u_t$. For polynomials f and g, define their truncated product by
$$f *_{\le d} g := \mathrm{tr}_{\le d}(fg),$$
where $\mathrm{tr}_{\le d}$ deletes all monomials of total degree greater than d. For every monomial M of degree at most d,
$$\mathrm{coeff}_M(f *_{\le d} g) = \sum_{M_1 M_2 = M} \mathrm{coeff}_{M_1}(f)\,\mathrm{coeff}_{M_2}(g),$$
where the sum ranges over all pairs of monomials (M_1, M_2) whose product is M. If $M = x_1^{a_1} \cdots x_q^{a_q}$, then the number of such pairs is
$$\prod_{j=1}^{q} (a_j + 1) \le 2^{\sum_j a_j} = 2^{\deg M} \le 2^d.$$
Hence the coefficient of every truncated product can be computed using only polynomially many arithmetic operations in N. For a product gate of arbitrary fan-in t, we compute
$$\mathrm{tr}_{\le d}(C_{u_1} \cdots C_{u_t})$$

by a balanced tree of truncated binary products. This requires $O(\log t)$ stages. Since t is bounded by the circuit size and the circuit encoding has length polynomially bounded in N, the resulting computation is in FNC^2.

The truncation does not affect coefficients of degree at most d. More precisely, for arbitrary polynomials f and g,

$$\mathrm{tr}_{\le d}(fg) = \mathrm{tr}_{\le d}(\mathrm{tr}_{\le d}(f)\,\mathrm{tr}_{\le d}(g)),$$

because every monomial occurring in a product has degree equal to the sum of the degrees of its factors. Thus a term of degree greater than d at an intermediate gate can never contribute to a coefficient of degree at most d at a later multiplication gate. The analogous statement for addition is immediate. Since the algebraic circuit has constant depth, we apply this procedure layer by layer. Addition of polynomially many encoded field elements, computation of a binary truncated product of coefficient vectors, and balanced iteration of truncated products are all computable in FNC^2. Hence $\mathbf{Coeff} \in \mathsf{FNC}^2$. Moreover, VNC^2 proves the correctness of the procedure:

$$\mathsf{VNC}^2 \vdash \mathbf{Coeff}(V, G, I, E, d, N) = \mathrm{coeff}_{\le d,N}(C).$$

If the polynomial computed by C has total degree at most d, this is its full coefficient vector, padded to N coordinates. All intermediate coefficient vectors have at most N coordinates and total encoding length polynomially bounded in N. All auxiliary indices and strings used by the computation have length polynomially bounded in N. Thus **Coeff** is represented by a function symbol in $\overline{\mathsf{VNC}^2}$ and may be used in $\Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})$ formulas.

5.3.7 Field-independent set-multilinearization

We now formalise the field-independent set-multilinearization argument of [For24]. Recall that N is our global size parameter and

$$n = n(N) \coloneqq \lfloor \log_2 N \rfloor \qquad \text{and} \qquad d = d(N) \coloneqq \lfloor c \log_2 n \rfloor,$$

where c is a fixed constant.

Let s be a fixed polynomial bounding the size of the original algebraic circuits. Since $s(n) = n^{O(1)}$ and $d = O(\log n)$, there is a fixed constant C such that $\binom{d\,s(n)+d}{d} \le (d\,s(n)+d)^d \le 2^{C(\log n)^2}$.

Since $C(\log n)^2 \le n$ for all sufficiently large n, there is a standard constant N_0 such that, for every $N \ge N_0$, $\binom{d\,s(n)+d}{d} \le 2^n \le N$. Thus all the coefficient vectors with degree at most d appearing are polynomially bounded by the coefficient-vector length N.

We will construct an FNC^2 function $\mathbf{set\text{-}multilinear}(V, G, I, E, X, d, N)$ which, given an algebraic circuit C encoded by (V, G, I, E) and a string X encoding a partition of the variables of C into d sets, outputs a set-multilinear circuit computing the set-multilinear projection of the polynomial computed by C. If C has product-depth Δ, the resulting circuit has product-depth at most 2Δ and size $\mathrm{poly}(s, d^d)$.

Definition 5.41. Let $X = (x_{i,j})$ be an $a \times m$ symbolic matrix of pairwise distinct variables, where $a \le m$. The *rectangular Permanent* is

$$\mathrm{perm}_{a\times m}(X) \coloneqq \sum_{\sigma:[a]\hookrightarrow[m]} x_{1,\sigma(1)} \cdots x_{a,\sigma(a)},$$

where the sum ranges over all injective maps $\sigma : [a] \hookrightarrow [m]$.

The polynomial $\mathrm{perm}_{a\times m}(X)$ is set-multilinear with respect to the partition of its variables into the a rows of X.

Theorem 5.42 (Binet–Minc Identity [Min79])**.** *Let $X = (x_{i,j})$ be an $a \times m$ symbolic matrix, where $a \le m$. Then*

$$\operatorname{perm}_{a\times m}(X) = \sum_{\mathcal{F}\in P_a} (-1)^{a-|\mathcal{F}|} \prod_{S\in\mathcal{F}} (|S|-1)! \left(\sum_{j=1}^{m} \prod_{i\in S} x_{i,j} \right), \tag{13}$$

where P_a denotes the set of all set partitions of $[a]$ into nonempty parts.

Every polynomial occurring in (13) has total degree at most a. In all applications below, we have $a \le d$ and $m \le s(n)$. Hence the number of variables is at most $d \cdot s(n)$, and by the choice of N_0, the coefficient vectors of degree at most d of both sides of (13) have length at most N. Therefore the standard coefficient-wise proof of the Binet–Minc identity can be formalised in VNC^2 using the function **Coeff** defined above.

Definition 5.43. For $k \ge 1$, define the Stirling number of the second kind by

$$S(a,k) = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^a.$$

We set $S(0,0) = 1$ and $S(a,0) = 0$ for $a > 0$. The ath Bell number is

$$B_a := \sum_{k=0}^{a} S(a,k).$$

Lemma 5.44. *For all integers $0 \le i < p$, $\sum_{j=0}^{p} (-1)^{p-j} \binom{p}{j} j^i = 0$. Consequently, $S(i,p) = 0$ whenever $p > i$.*

Proof. We prove the first identity by strong induction on i. For $i = 0$,

$$\sum_{j=0}^{p} (-1)^{p-j} \binom{p}{j} = (1-1)^p = 0$$

for every $p > 0$. Now let $i \ge 1$ and assume the statement holds for every exponent smaller than i. Let $p > i$. Since the $j = 0$ term vanishes,

$$\begin{aligned} \sum_{j=0}^{p} (-1)^{p-j} \binom{p}{j} j^i &= \sum_{j=1}^{p} (-1)^{p-j} \binom{p}{j} j^i \\ &= p \sum_{j=1}^{p} (-1)^{p-j} \binom{p-1}{j-1} j^{i-1}. \end{aligned}$$

Writing $q = j - 1$, this is

$$p \sum_{q=0}^{p-1} (-1)^{p-1-q} \binom{p-1}{q} (q+1)^{i-1}.$$

By the Binomial Theorem,

$$(q+1)^{i-1} = \sum_{r=0}^{i-1} \binom{i-1}{r} q^r.$$

Therefore the preceding expression equals

$$p \sum_{r=0}^{i-1} \binom{i-1}{r} \sum_{q=0}^{p-1} (-1)^{p-1-q} \binom{p-1}{q} q^r.$$

Since $p > i$, we have $p - 1 > r$ for every $r \le i - 1$. Each inner sum is therefore zero by the induction hypothesis. This proves the first identity. The conclusion about $S(i,p)$ follows immediately from its explicit formula. □

Lemma 5.45. VNC^2 *proves that for every* $a \geq 1$, $B_a \leq a^a$.

Proof. We use the recurrence $B_{a+1} = \sum_{i=0}^{a} \binom{a}{i} B_i$. We first derive it from the explicit formula for the Stirling numbers. For $k \geq 1$,

$$\begin{aligned} S(a+1,k) &= \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^{a+1} \\ &= \frac{1}{(k-1)!} \sum_{q=0}^{k-1} (-1)^{k-1-q} \binom{k-1}{q} (q+1)^a \\ &= \sum_{i=0}^{a} \binom{a}{i} S(i,k-1). \end{aligned}$$

Hence

$$B_{a+1} = \sum_{k=1}^{a+1} S(a+1,k) = \sum_{i=0}^{a} \binom{a}{i} \sum_{p=0}^{a} S(i,p).$$

By Lemma 5.44, $S(i,p) = 0$ whenever $p > i$, and therefore $B_{a+1} = \sum_{i=0}^{a} \binom{a}{i} B_i$.

We now prove $B_a \leq a^a$ by induction on a. The case $a = 1$ is immediate. Assume that $B_i \leq i^i$ for every $1 \leq i \leq a$. Using $B_0 = 1$ and the bound $i^i \leq a^i$ for $i \leq a$, we obtain

$$\begin{aligned} B_{a+1} &= \sum_{i=0}^{a} \binom{a}{i} B_i \\ &\leq \sum_{i=0}^{a} \binom{a}{i} a^i \\ &= (a+1)^a \\ &\leq (a+1)^{a+1}. \end{aligned}$$

This completes the induction. □

Thus we may use the estimate $B_a = O(a^a)$. This is weaker than the sharper estimate used in [For24], but it is sufficient for our purposes.

The Binet–Minc identity immediately gives the following circuit upper bound.

Corollary 5.46. *The rectangular Permanent* $\mathrm{perm}_{a\times m}$ *has a* $\sum^{B_a} \prod^{a} \sum^{m} \prod^{a}$ *formula of size* $\mathrm{poly}(m, a^a)$. *Moreover, the formula is set-multilinear with respect to the partition of the variables into rows.*

We next need a variant of the rectangular Permanent in which some specified columns are required to occur in the image of the injection.

Definition 5.47. Let $0 \leq k \leq a \leq m$. Define the *k-surjective rectangular Permanent* $\mathrm{perm}_{a\times m;k}(X)$ by

$$\mathrm{perm}_{a\times m;k}(X) = \sum_{\substack{\sigma:[a]\hookrightarrow[m] \\ \mathrm{im}(\sigma)\supseteq[k]}} x_{1,\sigma(1)} \cdots x_{a,\sigma(a)}.$$

Again, $\mathrm{perm}_{a\times m;k}$ is set-multilinear with respect to the partition into rows.

Lemma 5.48. *Let* $0 \leq k \leq a \leq m$. *Then*

$$\begin{aligned} \mathrm{perm}_{a\times m;k}(X) = \sum_{R\in\binom{[a]}{k}} &\mathrm{perm}_{k\times k}(X \restriction (R \times [k])) \\ &\cdot \mathrm{perm}_{(a-k)\times(m-k)}(X \restriction (([a]\setminus R) \times ([m]\setminus[k]))). \end{aligned}$$

This expression is set-multilinear with respect to the partition into rows.

Proof. Every injection $\sigma : [a] \hookrightarrow [m]$ whose image contains $[k]$ has a unique set $R = \sigma^{-1}([k])$ of size k. The restriction $\sigma|_R$ is a bijection from R to $[k]$, while the restriction to $[a] \setminus R$ is an injection into $[m] \setminus [k]$. Conversely, every such pair of maps determines a unique injection σ whose image contains $[k]$. Summing the corresponding monomials gives the identity. □

Corollary 5.49. *Let $0 \le k \le a \le m$. Then the k-surjective rectangular Permanent $\mathrm{perm}_{a\times m;k}$ has a depth-4 formula of size $\mathrm{poly}(m, a^a)$ which is set-multilinear with respect to its rows.*

Proof. By Lemma 5.48 and Corollary 5.46, for each $R \in \binom{[a]}{k}$ the two rectangular Permanents in the corresponding summand have formulas whose top addition gates have fan-in at most B_k and B_{a-k}, respectively. We distribute the product of these two formulas over their top addition gates. This gives at most $B_k B_{a-k}$ summands for each R; after merging adjacent multiplication gates, the resulting formula still has depth 4. Since there are $\binom{a}{k}$ choices for R, the resulting formula has size

$$\mathrm{poly}\left(m, \binom{a}{k} B_k B_{a-k}\right).$$

Using Lemma 5.45,

$$\binom{a}{k} B_k B_{a-k} \le 2^a k^k (a-k)^{a-k} \le 2^a a^a.$$

For $a \ge 2$, $2^a a^a = (2a)^a \le a^{2a}$, which is polynomial in a^a. The cases $a = 0, 1$ are immediate. Hence the formula has size $\mathrm{poly}(m, a^a)$. □

In the applications below, $a \le d$ and $m \le s(n)$. Therefore all circuits in the preceding construction have degree at most d, use at most $d \cdot s(n)$ variables, and have coefficient vectors of length at most N for $N \ge N_0$. Hence the preceding identities and circuit constructions can be verified in VNC^2 using **Coeff**.

Definition 5.50. Let the variables $\overline{x}$ be partitioned as $\overline{x} = \overline{x}_1 \sqcup \cdots \sqcup \overline{x}_d$. For $S \subseteq [d]$, a monomial is *S-set-multilinear* if it contains exactly one variable from $\overline{x}_i$ for each $i \in S$ and no variables from $\overline{x}_j$ for $j \notin S$. Define the *S-set-multilinear projection* $\pi_{sm,S}$ to be the linear map which fixes every S-set-multilinear monomial and sends every other monomial to zero.

A monomial is called *at-most-set-multilinear* if it is S-set-multilinear for some $S \subseteq [d]$. Define $\pi_{\neg sm}$ to be the linear map which sends every at-most-set-multilinear monomial to zero and fixes every other monomial.

For $S = [d]$, we write

$$\pi_{sm}(f) \coloneqq \pi_{sm,[d]}(f).$$

For $S = \emptyset$, $\pi_{sm,\emptyset}(f) = f(\overline{0})$. Every monomial belongs to exactly one of the following classes: it is S-set-multilinear for a unique $S \subseteq [d]$, or it is not at-most-set-multilinear. Hence

$$f = \pi_{\neg sm}(f) + \sum_{S \subseteq [d]} \pi_{sm,S}(f).$$

Addition is simulated by adding the corresponding set-multilinear parts.

Lemma 5.51. *Let $f = \alpha_1 f_1 + \cdots + \alpha_m f_m$. For every $S \subseteq [d]$, $\pi_{sm,S}(f) = \sum_{j=1}^m \alpha_j \pi_{sm,S}(f_j)$. Thus, the polynomial $\pi_{sm,S}(f)$ can be computed from the inputs $\{\pi_{sm,S}(f_j) : j \in [m]\}$ by a depth-1 set-multilinear sum circuit of size $O(m)$. Simultaneously maintaining all subsets $S \subseteq [d]$ incurs an additional factor 2^d.*

We now simulate multiplication gates.

Lemma 5.52. *Let $f = f_1 f_2 \cdots f_m$, and let $S \subseteq [d]$ with $|S| = \ell$. Then the polynomial $\pi_{sm,S}(f)$ can be computed from the inputs $\{\pi_{sm,T}(f_j) : T \subseteq [d],\ j \in [m]\}$ by a depth-4 set-multilinear circuit of size* $\mathrm{poly}(m, \ell^\ell, 2^d)$.

Proof. Rearrange the factors so that $f_1(\overline{0}) = \cdots = f_k(\overline{0}) = 0$ and $f_{k+1}(\overline{0}), \ldots, f_m(\overline{0}) \neq 0$ for some $0 \leq k \leq m$.

For $i > k$, define

$$g_i = \frac{f_i}{f_i(\overline{0})},$$

and for $i \leq k$, put $g_i = f_i$. Thus, for $i \leq k$, $g_i(\overline{0}) = 0$ whereas $g_i(\overline{0}) = 1$ for $i > k$. If $g = \prod_{i=1}^m g_i$, then

$$f = \left(\prod_{i>k} f_i(\overline{0})\right) g,$$

and therefore

$$\pi_{sm,S}(f) = \left(\prod_{i>k} f_i(\overline{0})\right) \pi_{sm,S}(g).$$

If $k > \ell$, then $\pi_{sm,S}(g) = 0$. Indeed, each of the first k factors has zero constant term, so a term contributing to an at-most-set-multilinear monomial must choose a nonconstant set-multilinear part from each of these k factors. These parts must involve pairwise disjoint nonempty subsets of S, which is impossible when $k > |S|$.

Assume that $k \leq \ell$ henceforth. We have

$$g = \prod_{i=1}^{k} \left(\pi_{\neg sm}(g_i) + \sum_{\emptyset \neq T \subseteq [d]} \pi_{sm,T}(g_i)\right) \cdot \prod_{i=k+1}^{m} \left(\pi_{\neg sm}(g_i) + 1 + \sum_{\emptyset \neq T \subseteq [d]} \pi_{sm,T}(g_i)\right).$$

Any product term containing a monomial from $\pi_{\neg sm}(g_i)$ cannot contribute to an at-most-set-multilinear monomial, since multiplying a non-at-most-set-multilinear monomial by another monomial cannot make it at-most-set-multilinear.

Therefore, a term contributing to $\pi_{sm,S}(g)$ is obtained by choosing pairwise disjoint nonempty subsets of S from distinct factors, whose union is S. Equivalently, these subsets form a set partition $\mathcal{F} \in P_S$. Since the first k factors have no constant term, each index in $[k]$ must be used.

Thus

$$\pi_{sm,S}(g) = \sum_{\substack{\mathcal{F} \in P_S \\ k \leq |\mathcal{F}| \leq m}} \sum_{\substack{\sigma : \mathcal{F} \hookrightarrow [m] \\ \mathrm{im}(\sigma) \supseteq [k]}} \prod_{T \in \mathcal{F}} \pi_{sm,T}(g_{\sigma(T)}).$$

For a partition $\mathcal{F}$ of S, define the $|\mathcal{F}| \times m$ matrix $A_{\mathcal{F}}$ by $(A_{\mathcal{F}})_{T,j} = \pi_{sm,T}(g_j)$, for $T \in \mathcal{F}$ and $j \in [m]$. Then

$$\pi_{sm,S}(g) = \sum_{\substack{\mathcal{F} \in P_S \\ k \leq |\mathcal{F}| \leq m}} \mathrm{perm}_{|\mathcal{F}| \times m; k}(A_{\mathcal{F}}).$$

Since $|\mathcal{F}| \leq |S| = \ell$, Corollary 5.49 gives a depth-4 set-multilinear circuit of size $\mathrm{poly}(m, \ell^\ell)$ for each summand. There are $B_\ell \leq \ell^\ell$ partitions of S. Hence $\pi_{sm,S}(g)$, and therefore $\pi_{sm,S}(f)$, has a depth-4 set-multilinear circuit of size $\mathrm{poly}(m, \ell^\ell, 2^d)$. □

For the formalisation, it is convenient to introduce the coefficient-vector version of the set-multilinear projection. Given a circuit C encoded by the set V, G, I, E, the partition X, and $S \subseteq [d(N)]$, define $\mathbf{SMCoeff}(V, G, I, E, X, S, N)$ to be the N-coordinate coefficient vector obtained from $\mathbf{Coeff}(V, G, I, E, d(N), N)$ by retaining the coefficient of a monomial precisely when that monomial is S-set-multilinear with respect to X, and replacing all other coefficients by zero.

Since decoding a monomial and checking whether it is S-set-multilinear are in FNC^2, the function **SMCoeff** is in FNC^2. Semantically,

$$\mathbf{SMCoeff}(V, G, I, E, X, S, N) = \mathrm{coeff}_{\leq d,N}(\pi_{sm,S}(C)).$$

For $S = [d(N)]$, we have

$$\mathbf{SMCoeff}(V, G, I, E, X, [d(N)], N) = \mathrm{coeff}_{\leq d,N}(\pi_{sm}(C)).$$

Applying Lemmas 5.51 and 5.52 gate by gate gives the field-independent set-multilinearization theorem.

Corollary 5.53 (Field-independent set-multilinearization [For24])**.** *Let $\mathbb{F}$ be a field. Let the variables $\overline{x}$ be partitioned as $\overline{x} = \overline{x}_1 \sqcup \cdots \sqcup \overline{x}_d$. Suppose that a polynomial $f \in \mathbb{F}[\overline{x}]$ is computed by a size-s algebraic circuit of product-depth Δ. Then $\pi_{sm}(f)$ is computed by a set-multilinear circuit of product-depth at most 2Δ and size* $\mathrm{poly}(s, d^d)$.

We now state the formal version using the global parameter N. Let s be the fixed standard polynomial bounding the size of the original circuit. There is a fixed standard polynomial s' and an FNC^2 function

$$\mathbf{set\text{-}multilinear}(V, G, I, E, X, d, N)$$

such that, for sufficiently large N, if $n = \lfloor \log_2 N \rfloor$ and $d = \lfloor c \log_2 n \rfloor$, and if (V, G, I, E) encodes a circuit C of size at most $s(n)$ and product-depth at most Δ, then

$$C' = \mathbf{set\text{-}multilinear}(V, G, I, E, X, d, N)$$

encodes a set-multilinear circuit satisfying C' computing $\pi_{sm}(C)$ with $\mathbf{Size}(C') < s'(s(n), d^d)$, and $\mathbf{ProdDepth}_{2\Delta}(C')$.

More precisely, working in the universal conservative extension $\overline{\mathsf{VNC}^2}$, we have

$$\begin{aligned}
\forall N \geq N_0\ \forall V, G, I, E, X\ \Big(&\mathbf{Circuit}(V, G, I, E) \wedge \mathbf{Size}(V, G, I, E) < s(n(N)) \\
&\wedge \mathbf{ProdDepth}_{\Delta}(V, G, I, E) \wedge \mathbf{Partition}(X, d(N)) \\
&\longrightarrow \mathbf{Coeff}(C', d(N), N) = \mathbf{SMCoeff}(V, G, I, E, X, [d(N)], N) \\
&\quad \wedge \mathbf{SetMultCircuit}(C', X, d(N)) \\
&\quad \wedge \mathbf{Size}(C') < s'(s(n(N)), d(N)^{d(N)}) \\
&\quad \wedge \mathbf{ProdDepth}_{2\Delta}(C')\Big),
\end{aligned}$$

where $C' = \mathbf{set\text{-}multilinear}(V, G, I, E, X, d(N), N)$. Here, $n(N)$ and $d(N)$ are the parameter functions fixed above, and N_0 is a fixed standard constant chosen so that all relevant encodings have length polynomially bounded in N. The predicate $\mathbf{Circuit}(V, G, I, E)$ checks that (V, G, I, E) is a valid encoding of an algebraic circuit, $\mathbf{Size}(V, G, I, E)$ returns its size, and $\mathbf{ProdDepth}_{\Delta}(V, G, I, E)$ checks that its product-depth is at most Δ. The predicate $\mathbf{Partition}(X, d)$ checks that X encodes a partition of the circuit variables into d sets. The function $\mathbf{Coeff}(C', d(N), N)$ outputs the degree-at-most-$d(N)$ coefficient vector of C', padded to length N, while $\mathbf{SMCoeff}(V, G, I, E, X, [d(N)], N)$ outputs the corresponding coefficient vector of the set-multilinear projection of the polynomial computed by (V, G, I, E). Finally, $\mathbf{SetMultCircuit}(C', X, d(N))$ checks that C' is set-multilinear with respect to the partition encoded by X. The functions s and s' are fixed standard polynomials giving, respectively, the assumed size bound on the original circuit and the size bound resulting from the set-multilinearization construction.

Since $2^{d(N)} = n(N)^{O(1)}$ and $d(N)^{d(N)} = 2^{O(\log n(N) \log\log n(N))} \leq N$ for all sufficiently large N, the size of C' and the lengths of all strings used in the construction are polynomially bounded in N.

The functions which enumerate subsets and set partitions, construct the Binet–Minc circuits, form the surjective-Permanent circuits, and assemble the gate-by-gate simulation are therefore in FNC^2.

Consequently, **set-multilinear** is represented by a function symbol in $\overline{\mathsf{VNC}^2}$, and the displayed correctness statement can be expressed by a $\Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})$ formula. By conservativity, the corresponding statement in the original language is provable in VNC^2.

5.3.8 Lower bounds for small-depth circuits

Recall that $n = n(N) \coloneqq \lfloor \log_2 N \rfloor$, $d = d(N) \coloneqq \lfloor c \log_2 n \rfloor$ and $k = k(N) \coloneqq 10d$. For the lower-bound construction, define $a(k) \coloneqq \max\{h \leq k : 2h^2 \leq k^2\}$, and let $A(k) \coloneqq \{a(k), -k\}$. Namely, $a(k) = \lfloor k/\sqrt{2} \rfloor$.

For every fixed standard integer $\delta \geq 1$, define

$$D_\delta(m) \coloneqq \max\{q \leq m : q^{2^\delta - 1} \leq m\}.$$

Thus $D_\delta(m) = \left\lfloor m^{1/(2^\delta-1)} \right\rfloor = \Theta\left(m^{1/(2^\delta-1)}\right)$. Since δ is a fixed standard constant, D_δ is an FNC^2 function.

We put $\epsilon_N \coloneqq 2^{-N}$ and $\eta_N \coloneqq \frac{1+\epsilon_N}{1-\epsilon_N}$. By the relative-rank properties proved in the previous section, VNC^2 proves that relative rank is imbalance, which is

$$\mathbf{relrank}(u, m, N, M_u(f)) \leq \frac{\eta_N}{P_\times(|s(u)|_1/2, N, N)},$$

and that if $f = f_1 \cdots f_t$ is a set-multilinear product with respect to a partition $(S_1, \ldots, S_t)$, then

$$\mathbf{relrank}(u, m, N, M_u(f)) \leq \eta_N^t \prod_{i=1}^{t} \mathbf{relrank}\left(u \restriction S_i, |S_i|, N, M_{u\restriction S_i}(f_i)\right).$$

We will also repeatedly use the following direct consequences of the approximation bounds. For $z_1, \ldots, z_t \geq 0$,

$$\prod_{i=1}^{t} \frac{1}{P_\times(z_i, N, N)} \leq \eta_N^t \frac{1}{P_\times(\sum_{i=1}^t z_i, N, N)},$$

and if $0 \leq z \leq z'$, then

$$\frac{1}{P_\times(z', N, N)} \leq \eta_N \frac{1}{P_\times(z, N, N)}.$$

Indeed, both inequalities follow directly from

$$(1 - \epsilon_N)2^z \leq P_\times(z, N, N) \leq (1 + \epsilon_N)2^z.$$

Lemma 5.54. *Let $w \in A(k)^d$ be any k-unbiased word, where $k \geq 10d$ and $d = \lfloor c \log n \rfloor$. Any set-multilinear circuit C of product-depth Δ computing P_w has size at least $n^{\Omega\left(\frac{D_\Delta(d)}{\Delta}\right)} = n^{\Omega\left(\frac{d^{1/(2^\Delta-1)}}{\Delta}\right)}$.*

We first prove the relative-rank upper bound for formulas. It is important that the formal statement is uniform in the degree m, since the induction will be applied to factors of degree smaller than d.

Lemma 5.55. *Let $1 \leq m \leq d$ and let $u \in A(k)^m$, where $k \geq 10d$. For a constant $\Delta \geq 1$, any set-multilinear formula C of product-depth Δ, degree m, and size at most s satisfies*

$$\mathrm{relrk}_u(C) \leq s \cdot 2^{-\frac{kD_\Delta(m)}{20}}.$$

Moreover, VNC^2 proves the corresponding rational inequality

$$\mathbf{relrank}(u, m, N, M_u(C)) \leq \frac{s\eta_N^{5\Delta m}}{P_\times(kD_\Delta(m)/20, N, N)}.$$

Proof. We prove both statements simultaneously by induction on product depth Δ. If $\Delta = 1$, then $C = C_1 + \cdots + C_t$, where each C_i is a product of m linear forms

$$C_i = L_1 \cdots L_m.$$

Since every entry of u is either $a(k)$ or $-k$, and $a(k) \geq \frac{k}{2}$, we have

$$\mathrm{relrk}_u(C_i) = \prod_{j=1}^{m} \mathrm{relrk}_{u_j}(L_j) \leq \prod_{j=1}^{m} 2^{-|u_j|_1/2} \leq 2^{-km/4}.$$

By subadditivity,

$$\mathrm{relrk}_u(C) \leq s2^{-km/4} \leq s2^{-km/20}.$$

Since $D_1(m) = m$, this proves the first statement.

In the theory, multiplicativity and imbalance give

$$\begin{aligned}
\mathbf{relrank}(u, m, N, M_u(C_i)) &\leq \eta_N^m \prod_{j=1}^{m} \mathbf{relrank}(u_j, 1, N, M_{u_j}(L_j)) \\
&\leq \eta_N^{2m} \prod_{j=1}^{m} \frac{1}{P_\times(|u_j|_1/2, N, N)} \\
&\leq \frac{\eta_N^{3m}}{P_\times(\frac{1}{2}\sum_{j=1}^{m} |u_j|_1, N, N)}.
\end{aligned}$$

Since $|u_j|_1 \geq k/2$, $\frac{1}{2}\sum_{j=1}^{m} |u_j|_1 \geq \frac{km}{4}$. Hence

$$\mathbf{relrank}(u, m, N, M_u(C_i)) \leq \frac{\eta_N^{3m+1}}{P_\times(km/4, N, N)} \leq \frac{\eta_N^{3m+2}}{P_\times(km/20, N, N)} \leq \frac{\eta_N^{5m}}{P_\times(kD_1(m)/20, N, N)}.$$

By subadditivity,

$$\mathbf{relrank}(u, m, N, M_u(C)) \leq \frac{s\eta_N^{5m}}{P_\times(kD_1(m)/20, N, N)}.$$

Now assume the result holds for product-depth at most Δ, and let C have product-depth $\Delta + 1$. Write

$$C = C_1 + \cdots + C_t.$$

Put $q \coloneqq D_{\Delta+1}(m)$ and $T_\Delta \coloneqq q^{2^\Delta - 1}$. By the definition of $D_{\Delta+1}$, $q^{2^{\Delta+1}-1} \leq m$. Therefore $qT_\Delta^2 = q^{2^{\Delta+1}-1} \leq m$, and in particular

$$T_\Delta^2 \leq m \leq d.$$

Each C_i of size s_i is said to be of type 1 if one of its factors has degree at least T_Δ, and otherwise it is of type 2.

- **Type 1.** Suppose

  $$C_i = C_{i,1} \cdots C_{i,t_i},$$

  and, after reordering, let $m' \coloneqq \deg(C_{i,1}) \geq T_\Delta$. Let $S_{i,1}$ be the set of variable blocks associated with $C_{i,1}$, so $|S_{i,1}| = m'$. Since $q^{2^\Delta - 1} = T_\Delta \leq m'$, we have $D_\Delta(m') \geq q$. The induction hypothesis therefore gives

  $$\begin{aligned}
  \mathrm{relrk}_u(C_i) &\leq \mathrm{relrk}_{u\restriction S_{i,1}}(C_{i,1}) \\
  &\leq s_{i,1} 2^{-kD_\Delta(m')/20} \\
  &\leq s_i 2^{-kq/20}.
  \end{aligned}$$

We now prove the corresponding statement in the theory. Constant factors may be absorbed into one of the nonconstant factors; if one factor is zero, then $C_i = 0$. Thus we may assume that every factor has positive degree, and consequently $t_i \le m$.

For every set-multilinear polynomial f we have

$$\mathbf{relrank}(v, r, N, M_v(f)) \le \frac{1}{1-\epsilon_N} \le \eta_N,$$

since $\mathbf{rank}(M_v(f)) \le 2^{L(v)/2}$.

Hence approximate multiplicativity gives

$$\begin{aligned}\mathbf{relrank}(u, m, N, M_u(C_i)) &\le \eta_N^{t_i} \prod_{j=1}^{t_i} \mathbf{relrank}\left(u \restriction S_{i,j}, |S_{i,j}|, N, M_{u\restriction S_{i,j}}(C_{i,j})\right) \\ &\le \eta_N^{2m} \mathbf{relrank}\left(u \restriction S_{i,1}, m', N, M_{u\restriction S_{i,1}}(C_{i,1})\right).\end{aligned}$$

Applying the induction hypothesis,

$$\mathbf{relrank}(u, m, N, M_u(C_i)) \le \frac{s_{i,1}\eta_N^{2m+5\Delta m'}}{P_\times(kD_\Delta(m')/20, N, N)}.$$

Since $m' \le m$ and $D_\Delta(m') \ge q$,

$$\mathbf{relrank}(u, m, N, M_u(C_i)) \le \frac{s_i\eta_N^{2m+5\Delta m+1}}{P_\times(kq/20, N, N)} \le \frac{s_i\eta_N^{5(\Delta+1)m}}{P_\times(kq/20, N, N)}.$$

- **Type 2.** Suppose

$$C_i = C_{i,1} \cdots C_{i,t_i},$$

where each nonconstant factor $C_{i,j}$ has degree $\deg(C_{i,j}) < T_\Delta$. As above, we absorb all nonzero constant factors. Since C_i is set-multilinear of degree $m = \sum_{j=1}^{t_i} \deg(C_{i,j})$. Therefore $t_i T_\Delta > m$, and hence $t_i > \frac{m}{T_\Delta}$.

As the formula is set-multilinear, $(S_{i,1}, \ldots, S_{i,t_i})$ forms a partition of $[m]$, where $C_{i,j}$ is set-multilinear with respect to the variable blocks indexed by $S_{i,j}$. Let $u^{ij} \coloneqq u \restriction S_{i,j}$. Since $C_{i,j}$ is set-multilinear, we have

$$|u^{ij}| = |S_{i,j}| = \deg(C_{i,j}).$$

Let a_{ij} be the number of entries of u^{ij} equal to $a(k)$, and let b_{ij} be the number of entries equal to $-k$. Then

$$a_{ij} + b_{ij} = |S_{i,j}| = \deg(C_{i,j}),$$

so in particular $b_{ij} = \deg(C_{i,j}) - a_{ij}$.

Therefore, we have

$$s(u^{ij}) \coloneqq \sum_{\ell \in S_{i,j}} u_\ell = a_{ij}a(k) - b_{ij}k.$$

We use the following elementary Diophantine estimate.

Claim 5.56. Let $T \ge 1$ and let $a, b \ge 0$ satisfy $1 \le a + b < T$. If $k \ge 10T^2$, then

$$|aa(k) - bk|_1 \ge \frac{k}{10T}.$$

Proof. Put $h = a(k)$. By the definition of h, we have

$$2h^2 \le k^2 < 2(h+1)^2,$$

and therefore

$$0 \le k^2 - 2h^2 < 4h + 2 \le 5k.$$

We first note that

$$a^2 \neq 2b^2$$

whenever $(a, b) \neq (0, 0)$. Indeed, the equation $a^2 = 2b^2$ has no nonzero integer solution. This follows by the standard parity descent: a must be even, and then b must also be even, producing a strictly smaller solution. The same argument is formalizable in V^0 by bounded induction.

Suppose, for contradiction, that

$$10T|ah - bk|_1 < k.$$

Put $\delta \coloneqq ah - bk$. Since $a^2 - 2b^2$ is a nonzero integer,

$$k^2 \le |(a^2 - 2b^2)k^2|_1.$$

On the other hand,

$$\begin{aligned} |(a^2 - 2b^2)k^2|_1 &= \left|a^2(k^2 - 2h^2) + 2(a^2h^2 - b^2k^2)\right|_1 \\ &= \left|a^2(k^2 - 2h^2) + 2\delta(ah + bk)\right|_1 \\ &\le a^2|k^2 - 2h^2|_1 + 2|\delta|_1|ah + bk|_1. \end{aligned}$$

Since $a < T$ and $a + b < T$, we have

$$a^2|k^2 - 2h^2|_1 < 5T^2k \le \frac{k^2}{2},$$

while

$$|ah + bk|_1 < Tk.$$

Therefore

$$2|\delta|_1|ah + bk|_1 < 2\frac{k}{10T}Tk = \frac{k^2}{5}.$$

Hence

$$|(a^2 - 2b^2)k^2|_1 < \frac{7}{10}k^2,$$

which contradicts

$$|(a^2 - 2b^2)k^2|_1 \ge k^2.$$

□

Since

$$a_{ij} + b_{ij} = \deg(C_{i,j}) < T_\Delta$$

and

$$k \ge 10d \ge 10T_\Delta^2,$$

the claim gives

$$|s(u^{ij})|_1 \ge \frac{k}{10T_\Delta}.$$

Therefore, using multiplicativity and imbalance, we have

$$\begin{aligned}\mathrm{relrk}_u(C_i) &= \prod_{j=1}^{t_i} \mathrm{relrk}_{u^{ij}}(C_{i,j}) \\ &\le \prod_{j=1}^{t_i} 2^{-|s(u^{ij})|_1/2} \\ &\le 2^{-kt_i/(20T_\Delta)} \le 2^{-km/(20T_\Delta^2)} \le 2^{-kq/20},\end{aligned}$$

where the last inequality follows from $qT_\Delta^2 \le m$.

In the theory,

$$\begin{aligned}\mathbf{relrank}(u, m, N, M_u(C_i)) &\le \eta_N^{t_i} \prod_{j=1}^{t_i} \mathbf{relrank}\left(u^{ij}, |S_{i,j}|, N, M_{u^{ij}}(C_{i,j})\right) \\ &\le \eta_N^{2t_i} \prod_{j=1}^{t_i} \frac{1}{P_\times(|s(u^{ij})|_1/2, N, N)} \\ &\le \frac{\eta_N^{3t_i}}{P_\times(\frac{1}{2}\sum_{j=1}^{t_i} |s(u^{ij})|_1, N, N)}.\end{aligned}$$

Since

$$\frac{1}{2}\sum_{j=1}^{t_i} |s(u^{ij})|_1 \ge \frac{kt_i}{20T_\Delta},$$

we obtain

$$\mathbf{relrank}(u, m, N, M_u(C_i)) \le \frac{\eta_N^{3t_i+1}}{P_\times(kt_i/(20T_\Delta), N, N)}.$$

Moreover,

$$\frac{kt_i}{20T_\Delta} > \frac{km}{20T_\Delta^2} \ge \frac{kq}{20}.$$

Hence

$$\mathbf{relrank}(u, m, N, M_u(C_i)) \le \frac{\eta_N^{3t_i+2}}{P_\times(kq/20, N, N)} \le \frac{\eta_N^{5(\Delta+1)m}}{P_\times(kq/20, N, N)}.$$

Finally, by subadditivity,

$$\mathrm{relrk}_u(C) \le s2^{-kq/20},$$

and

$$\mathbf{relrank}(u, m, N, M_u(C)) \le \frac{s\eta_N^{5(\Delta+1)m}}{P_\times(kq/20, N, N)}.$$

Since $q = D_{\Delta+1}(m)$, this completes the induction. □

We now prove Lemma 5.54.

Proof of Lemma 5.54. Let C be a size-s set-multilinear circuit of product-depth Δ computing P_w. Unfolding C into a formula gives a set-multilinear formula of the same product-depth and size at most $s^{2\Delta}$. Since w is k-unbiased, we have $|s(w)|_1 \le k$. Since the coefficient matrix of the word polynomial has high rank, we have

$$\mathrm{relrk}_w(P_w) = 2^{-|s(w)|_1/2} \ge 2^{-k/2}.$$

Applying Lemma 5.55 gives $2^{-k/2} \le s^{2\Delta}2^{-kD_\Delta(d)/20}$.

Therefore, we have $s^{2\Delta} \geq 2^{k(D_\Delta(d)/20-1/2)}$. For all sufficiently large N, $D_\Delta(d) \geq 20$, and hence $s \geq 2^{\Omega\left(\frac{kD_\Delta(d)}{\Delta}\right)}$. Since $k = 10d = \Theta(\log n)$, we obtain

$$s \geq n^{\Omega\left(\frac{D_\Delta(d)}{\Delta}\right)} = n^{\Omega\left(\frac{d^{1/(2^\Delta-1)}}{\Delta}\right)}.$$

We now give the corresponding formal argument in VNC^2. From the high-rank statement proved in the preceding section, we have

$$\mathbf{relrank}\,(w, d, N, M_w(\mathbf{WordP}(w, d, N))) \geq \frac{\eta_N^{-1}}{P_\times(k/2, N, N)}.$$

Applying the formal part of Lemma 5.55 to the formula, we have

$$\mathbf{relrank}\,(w, d, N, M_w(\mathbf{WordP}(w, d, N))) \leq \frac{s^{2\Delta}\eta_N^{5\Delta d}}{P_\times(kD_\Delta(d)/20, N, N)}.$$

Consequently,

$$s^{2\Delta} \geq \eta_N^{-(5\Delta d+1)} \frac{P_\times(kD_\Delta(d)/20, N, N)}{P_\times(k/2, N, N)}.$$

By the approximation property of $P_\times$, we have

$$\frac{P_\times(kD_\Delta(d)/20, N, N)}{P_\times(k/2, N, N)} \geq \eta_N^{-1} 2^{k(D_\Delta(d)/20-1/2)}.$$

Hence

$$s^{2\Delta} \geq \eta_N^{-(5\Delta d+2)} 2^{k(D_\Delta(d)/20-1/2)}.$$

Since $\eta_N = \frac{1+2^{-N}}{1-2^{-N}}$ and $d = O(\log\log N)$, there is a fixed standard constant N_1 such that VNC^2 proves, for every $N \geq N_1$,

$$\eta_N^{5\Delta d+2} \leq 2.$$

For example, for sufficiently large N we have $\eta_N \leq 1 + 3 \cdot 2^{-N}$ and $6(5\Delta d + 2) \leq 2^N$. The desired estimate then follows from the Binomial Theorem. Thus the rational approximation loss does not affect the asymptotic lower bound. □

Finally, we combine the set-multilinear lower bound with the field-independent set-multilinearization theorem from the previous subsection.

Let $s(n)$ be a fixed polynomial. Suppose that a circuit C of size at most $s(n)$, depth at most 2Δ, and product-depth at most Δ satisfies

$$\mathbf{Coeff}(C, d(N), N) = \mathbf{WordP}(w, d, N).$$

Since $\mathbf{WordP}(w, d, N)$ is supported on the $[d]$-set-multilinear monomials, it follows that $\pi_{sm}(C) = P_w$.

Indeed, equality of the truncated coefficient vectors implies equality of all $[d]$-set-multilinear coefficients, and every such monomial has degree exactly d.

By the field-independent set-multilinearization theorem, there is a set-multilinear circuit C' computing P_w, of product-depth at most 2Δ, and size $\mathrm{poly}(s(n), d^d)$.

Since $s(n)$ is polynomial and $d = \Theta(\log n)$, we have

$$\mathrm{poly}(s(n), d^d) = n^{O(\log\log n)}.$$

On the other hand, Lemma 5.54, applied with product-depth 2Δ, gives

$$\mathbf{Size}(C') \geq n^{\Omega\left(\frac{D_{2\Delta}(d)}{2\Delta}\right)}.$$

For fixed Δ,

$$D_{2\Delta}(d) = \Theta\left(d^{1/(2^{2\Delta}-1)}\right),$$

and therefore

$$D_{2\Delta}(d) \gg \log\log n.$$

Hence these two bounds are incompatible for all sufficiently large N.

Consequently, there is a fixed standard constant N_2 such that $\overline{\mathsf{VNC}^2}$ proves that, when $s(n)$ is any fixed polynomial,

$$\begin{aligned}\forall N \geq N_2\, \forall w, V, G, I, E\, \Big(& w \in A(k(N))^{d(N)} \wedge \mathbf{unbiased}(w, d(N), k(N)) \\ & \wedge \mathbf{Circuit}_w(V, G, I, E) \wedge \mathbf{Size}(V, G, I, E) < s(n(N)) \\ & \wedge \mathbf{Depth}_{2\Delta}(V, G, I, E) \wedge \mathbf{ProdDepth}_{\Delta}(V, G, I, E) \\ & \longrightarrow \mathbf{Coeff}(V, G, I, E, d(N), N) \neq \mathbf{WordP}(w, d(N), N)\Big).\end{aligned}$$

Here $\mathbf{Circuit}_w(V, G, I, E)$ means that (V, G, I, E) is a valid algebraic circuit whose variables are the variables in $\overline{X}(w)$, and **Coeff** and **WordP** use the same fixed N-coordinate monomial encoding.

All number and string parameters occurring above have encodings of length polynomially bounded in N. Since Δ and c are fixed standard constants, the functions $n(N)$, $d(N)$, $k(N)$, $a(k)$, and D_δ are in FNC^2. Hence the matrix of the final displayed statement is a $\Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})$ formula in $\overline{\mathsf{VNC}^2}$, and by conservativity the corresponding statement is provable in VNC^2.

5.3.9 Lower bounds for permanent

Now, we show that VNC^2 is able to prove superpolynomial lower bounds for the Permanent polynomial against constant-depth algebraic circuits. We show that each word polynomial is a p-projection of a Permanent polynomial and that the projection can be generated uniformly and efficiently.

First, we recall the construction from [LST25]: for every b-unbiased word $w \in A^d$, an algebraic branching program (ABP) of width 2^b computing P_w can be constructed explicitly and efficiently. Here, we give the proof sketch in [LST25] to keep the exposition self-contained.

Lemma 5.57 ([LST25])**.** *Let $w \in A^d$ be any word that is b-unbiased. Then there is a set-multilinear ABP of width 2^b computing $P_w \in \mathbb{F}_{sm}[\overline{X}(w)]$. Moreover,*

$$\mathrm{relrk}_w(P_w) \geq 2^{-b/2}.$$

Proof. At each layer $i \in \{0, \ldots, d\}$, before identifying the vertices in the last layer, the ABP has exactly $2^{|w_{[i]}|_1} \leq 2^b$ vertices. Assume that $w_{[i]} \geq 0$. Then, for any monomial m in variable sets $X_1, \ldots, X_i$, the string $\sigma(m_+)$ has length exactly $w_{[i]}$ more than that of $\sigma(m_-)$. Assuming that $\sigma(m_+)$ agrees with $\sigma(m_-)$ on all but its last $w_{[i]}$ bits, i.e., $\sigma(m_+) = \sigma(m_-)\tau$ for $\tau \in \{0,1\}^{w_{[i]}}$, the vertex of the ABP keeps track of the string τ. More formally, for each $\tau \in \{0,1\}^{|w_{[i]}|}$ we have a vertex v_τ in the ith layer of the ABP, where the polynomial P_{v_τ} computed from the source node to v_τ is the sum over all monomials m over $X_1, \ldots, X_i$ such that $\sigma(m_+) = \sigma(m_-)\tau$ if $w_{[i]} \geq 0$, and $\sigma(m_-) = \sigma(m_+)\tau$ if $w_{[i]} < 0$.

Given a vertex u_τ on layer $i+1$, we have

$$P_{u_\tau} = \sum_{v_\rho} P_{v_\rho} \cdot L_{\rho,\tau}$$

for a suitable linear polynomial $L_{\rho,\tau} \in \mathbb{F}[X_{i+1}]$, where the sum runs over all vertices v_ρ in the ith layer. More precisely,

$$L_{\rho,\tau} = \begin{cases} 0 & \text{if } \operatorname{sgn}(w_{[i+1]}) = \operatorname{sgn}(w_{[i]}),\ |w_{[i+1]}| \geq |w_{[i]}|, \text{ and } \rho \text{ is not a prefix of } \tau, \\ x_{\rho'} & \text{if } \operatorname{sgn}(w_{[i+1]}) = \operatorname{sgn}(w_{[i]}),\ |w_{[i+1]}| \geq |w_{[i]}|, \text{ and } \tau = \rho\rho', \\ 0 & \text{if } \operatorname{sgn}(w_{[i+1]}) = \operatorname{sgn}(w_{[i]}),\ |w_{[i+1]}| < |w_{[i]}|, \text{ and } \tau \text{ is not a suffix of } \rho, \\ x_{\tau'} & \text{if } \operatorname{sgn}(w_{[i+1]}) = \operatorname{sgn}(w_{[i]}),\ |w_{[i+1]}| < |w_{[i]}|, \text{ and } \rho = \tau'\tau, \\ x_{\rho\tau} & \text{if } \operatorname{sgn}(w_{[i+1]}) \neq \operatorname{sgn}(w_{[i]}). \end{cases}$$

Here each x_γ denotes the variable of X_{i+1} labelled by the Boolean string γ.

Finally, we identify all the vertices in layer d into a single sink. The polynomial computed from the source to this sink is exactly P_w.

The high-relative-rank property

$$\operatorname{relrk}_w(P_w) \geq 2^{-b/2}$$

was proved in the preceding subsection. □

In the parameter regime used below, where $b = k(N)$ and 2^b is polynomially bounded in N, the ABP and its edge labels can be generated uniformly by an FNC^2 function from w, d, and N.

Let

$$m(w) \coloneqq 2 + \sum_{i=1}^{d-1} 2^{|w_{[i]}|_1}.$$

After identifying the last layer, the ABP constructed above has exactly $m(w)$ vertices. Since w is b-unbiased,

$$m(w) \leq 2 + (d-1)2^b.$$

Let $G = (V, E)$ be an edge-weighted graph. A cycle cover C of G is a set of node-disjoint directed cycles which contains every vertex of G. The weight $w(C)$ of C is the product of the weights of the edges in C. Then the Permanent of the weighted adjacency matrix M_G is equal to the sum of the weights of the cycle covers of G, i.e.,

$$\operatorname{perm}(M_G) = \sum_{\text{cycle cover } C \text{ of } G} w(C).$$

An ABP with source s and sink t can be converted into a graph G such that

$$\operatorname{perm}(M_G) = \sum_{\pi : s \rightsquigarrow t} \prod_{e \in \pi} L_e,$$

where L_e is the label of the edge e in the directed path π from s to t. We add an edge from t to s and label it by 1. Also, for each node v except s, we add a self-loop $v \to v$ and label it by 1. The adjacency matrix M_G is therefore

$$M_G(i,j) \coloneqq \begin{cases} \ell(v_i, v_j) & \text{if } v_i \to v_j \text{ is an edge of the ABP constructed above,} \\ 1 & \text{if } i = j \text{ and } v_i \neq s, \\ 1 & \text{if } (v_i, v_j) = (t, s), \\ 0 & \text{otherwise.} \end{cases}$$

Indeed, since the ABP is acyclic, every cycle cover of G must contain the edge $t \to s$. The remaining non-loop edges in the cycle cover therefore form a unique directed path from s to t, while every vertex outside this path is covered by its self-loop. Thus the cycle covers of G are in bijection with the source-to-sink paths in the ABP.

Let $Y = (y_{i,j})_{i,j\in[m(w)]}$ be the variables of the $m(w) \times m(w)$ Permanent. Define the projection ρ_w by replacing $y_{i,j}$ with the corresponding entry $M_G(i,j)$. Every entry of M_G is either a variable of $\overline{X}(w)$ or one of the constants $0, 1$. Hence ρ_w is a p-projection, and

$$\rho_w\Big(\mathrm{perm}_{m(w)}(Y)\Big) = P_w.$$

The projection ρ_w can be generated by an FNC^2 function. We also define an FNC^2 function

$$\mathbf{ProjCirc}(V, G, I, E, \rho_w)$$

which, given an algebraic circuit C in the variables $(y_{i,j})_{i,j\in[m]}$, replaces every input variable according to ρ_w. This substitution does not increase the size or the depth of the circuit. If C computes perm_m, then $\mathbf{ProjCirc}(C, \rho_w)$ computes P_w.

Recall that

$$n = n(N) \coloneqq \lfloor \log_2 N \rfloor, \qquad d = d(N) \coloneqq \lfloor c \log_2 n \rfloor, \qquad k = k(N) \coloneqq 10d,$$

where we choose the fixed constant c so that $0 < c < 1/20$. Let

$$m(N) \coloneqq 2 + d(N)2^{k(N)}.$$

For every $k(N)$-unbiased word $w \in A(k(N))^{d(N)}$, the ABP constructed above has at most $m(N)$ vertices. By adding isolated vertices with self-loops of weight 1, we may therefore regard P_w as a p-projection of the Permanent polynomial $\mathrm{perm}_{m(N)}$.

Since $d(N) = O(\log n(N))$ and $2^{k(N)} \le n(N)^{10c}$, we have

$$m(N)^2 = O\big(n(N)^{20c}(\log n(N))^2\big).$$

Thus the ABP, the matrix M_G, and the projection ρ_w all have encodings of length polynomially bounded in N. Moreover, for every circuit of size polynomially bounded in $m(N)$, the projected circuit also has encoding length polynomially bounded in N. The stronger assumption $20c < 1$ will be used below to ensure that the relevant coefficient spaces fit within the ambient parameter N.

Moreover, we define a predicate

$$\mathbf{SetMultOutput}(V, G, I, E, m, N)$$

which expresses that the polynomial computed by the circuit encoded by (V, G, I, E) is set-multilinear with respect to the row partition

$$Y_i = \{y_{i,1}, \ldots, y_{i,m}\}, \qquad i \in [m].$$

For the constant-depth, polynomial-size circuits considered below, we can compute a polynomial upper bound D_C on the syntactic degree of the output polynomial. Since the circuit has m^2 variables and $m = m(N) = O(n(N)^{10c} \log n(N))$, the assumption $20c < 1$ implies that, for sufficiently large N,

$$\binom{m^2 + D_C}{D_C} \le N.$$

Hence $\mathbf{Coeff}(V, G, I, E, D_C, N)$ computes the full coefficient vector of the output polynomial. The predicate **SetMultOutput** checks that every coefficient corresponding to a monomial which is not set-multilinear with respect to the row partition is zero. Thus **SetMultOutput** is a $\Sigma_0^B(\mathcal{L}_{\mathsf{FNC}^2})$ predicate.

Consequently,

$$\mathbf{SetMultOutput}(V, G, I, E, m(N), N) \wedge \mathbf{Coeff}(V, G, I, E, m(N), N) = \mathbf{Perm}(m(N), N)$$

implies that the polynomial computed by (V, G, I, E) is exactly $\mathrm{perm}_{m(N)}$. Indeed, if the output polynomial is set-multilinear with respect to the $m(N)$ row blocks, then every monomial occurring in it has total degree exactly $m(N)$. Hence $\mathbf{Coeff}(V, G, I, E, m(N), N)$ is its full coefficient vector, and equality with $\mathbf{Perm}(m(N), N)$ implies equality of the two polynomials.

Now suppose, for contradiction, that both

$$\mathbf{SetMultOutput}(V, G, I, E, m(N), N)$$

and

$$\mathbf{Coeff}(V, G, I, E, m(N), N) = \mathbf{Perm}(m(N), N)$$

hold. By the preceding observation, the circuit C encoded by (V, G, I, E) computes $\mathrm{perm}_{m(N)}$. Therefore $\mathbf{ProjCirc}(C, \rho_w)$ has the same size and depth as C and computes

$$\rho_w(\mathrm{perm}_{m(N)}) = P_w,$$

contradicting the lower bound for P_w.

Theorem 5.58 (Provability of superpolynomial lower bound for the Permanent polynomial against constant-depth algebraic circuits)**.** *Let s be a polynomial and let Δ be a constant. There is a fixed standard constant N_1 such that the following formula is provable in VNC^2:*

$$\begin{aligned}\forall N \geq N_1\, \forall V, G, I, E\, \Big(&\mathbf{Circuit}_{\mathrm{perm}}(V, G, I, E, m(N)) \\ &\wedge \mathbf{SetMultOutput}(V, G, I, E, m(N), N) \\ &\wedge \mathbf{Size}(V, G, I, E) < s(m(N)) \\ &\wedge \mathbf{Depth}_{2\Delta}(V, G, I, E) \\ &\longrightarrow \mathbf{Coeff}(V, G, I, E, m(N), N) \neq \mathbf{Perm}(m(N), N)\Big).\end{aligned}$$

The matrix of the displayed formula is a $\Sigma^B_0(\mathcal{L}_{\mathsf{FNC}^2})$ formula. Every string occurring in the proof has length polynomially bounded in N. Hence the Cook–Nguyen translation applies. By conservativity, the corresponding statement is provable in VNC^2.

Using the translation between bounded arithmetic theories and propositional proofs as shown in [CN10], we obtain the following theorem.

Theorem 5.59. *For every fixed polynomial s and every fixed constant Δ, the propositional translations asserting that the Permanent polynomial does not admit algebraic circuits of size at most $s(m)$ and depth at most 2Δ have polynomial-size NC^2-Frege proofs over the fixed coefficient field.*

Acknowledgements

We are indebted to the anonymous reviewers of this paper for many important comments.

AI notice

AI tools were not used in this work, except to improve the exposition of some parts of Section 5 and Table 1, and to suggest local fixes once the manuscript had been completed.

References

[ABSRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211 (electronic), 2002. 13

[AR23] Per Austrin and Kilian Risse. Sum-of-squares lower bounds for the minimum circuit size problem. In *38th Computational Complexity Conference, CCC'23*, pages 31:1–31:21, 2023. 6

[BCS97] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, Berlin, 1997. 3

[BDS24] CS Bhargav, Sagnik Dutta, and Nitin Saxena. Improved lower bound, and proof barrier, for constant depth algebraic circuits. *ACM Transactions on Computation Theory*, 16(4):1–22, 2024. 11

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizations of the P =? NP question. *SIAM J. Comput.*, 4(4):431–442, 1975. 2

[CKR+20] Prerona Chatterjee, Mrinal Kumar, C. Ramya, Ramprasad Saptharishi, and Anamay Tengse. On the existence of algebraically natural proofs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS '20*, pages 870–880. IEEE, 2020. 3

[CLO26] Lijie Chen, Jiatu Li, and Igor Oliveira. Reverse mathematics of complexity lower bounds. *SIAM Journal on Computing*, pages FOCS24–316, 2026. 2

[CN10] Stephen Cook and Phuong Nguyen. *Logical Foundations of Proof Complexity*. ASL Perspectives in Logic. Cambridge University Press, 2010. 4, 5, 25, 29, 30, 31, 58

[EGdOW18] Klim Efremenko, Ankit Garg, Rafael Mendes de Oliveira, and Avi Wigderson. Barriers for rank methods in arithmetic complexity. In *9th Innovations in Theoretical Computer Science Conference, ITCS'18*, pages 1:1–1:19, 2018. 1, 3

[For24] Michael A Forbes. Low-depth algebraic circuit lower bounds over any field. In *39th Computational Complexity Conference (CCC 2024)*, pages 31–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024. 1, 3, 4, 5, 6, 11, 24, 33, 42, 43, 45, 48

[FSTW21] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. *Theory Comput.*, 17:1–88, 2021. 14

[FSV18] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14(1):1–45, 2018. 3

[GGL+26] Michal Garlík, Svyatoslav Gryaznov, Jiaqi Lu, Rahul Santhanam, and Iddo Tzameret. Meta-Mathematics of Algebraic Complexity. In Claudia Faggian and Joost-Pieter Katoen, editors, *41st Annual Symposium on Logic in Computer Science (LICS 2026)*, volume 380 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. https://doi.org/10.4230/LIPIcs.LICS.2026.49. 1

[GGRT26] Michal Garlík, Svyatoslav Gryaznov, Hanlin Ren, and Iddo Tzameret. The weak rank principle: Lower bounds and applications. In Aditya Bhaskara and Artur Czumaj, editors, *Proceedings of the 58th Annual ACM Symposium on Theory of Computing, STOC 2026, Salt Lake City, UT, USA, June 22-26, 2026*, pages 138–149. ACM, 2026. 6, 8, 16, 17, 21

[GKSS17] Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017. 3

[GMdOW19] Ankit Garg, Visu Makam, Rafael Mendes de Oliveira, and Avi Wigderson. More barriers for rank methods, via a "numeric to symbolic" transfer. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS '19*, pages 824–844, 2019. 1, 3

[GP18] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018. 7, 14

[HJ18] J. Hamkins and E. Jeřabek. can-the-induction-axiom-in-the-peano-arithmetic-be-replaced-by-the-irrationality. Mathoverflow, 2018. 6

[HT15] Pavel Hrubes and Iddo Tzameret. Short proofs for the determinant identities. *SIAM J. Comput.*, 44(2):340–383, 2015. 14, 24

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999. 21

[Jeř22] Emil Jeřábek. Iterated multiplication in VTC^0. *Archive for Mathematical Logic*, 61(5):705–767, 2022. 32

[Jeř23] Emil Jeřábek. Elementary analytic functions in VTC^0. *Annals of Pure and Applied Logic*, 174(6):103269, 2023. 6, 32, 34, 35, 36

[KK23] Eitetsu Ken and Satoru Kuroda. On matrix rank function over bounded arithmetics. *arXiv preprint arXiv:2310.05982*, 2023. 6, 33, 39

[Kra95] Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*, volume 60 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1995. 4

[Kra04] Jan Krajíček. Dual weak pigeonhole principles, pseudo-surjective functions, and provability of circuit lower bounds. *Journal of Symbolic Logic*, 69(1):265–286, 2004. 2, 6

[Kra11] Jan Krajíček. On the proof complexity of the Nisan-Wigderson generator based on a hard $\mathrm{NP} \cap \mathrm{coNP}$ function. *J. Math. Log.*, 11(1):11–27, 2011. 2

[KRST22] Mrinal Kumar, C. Ramya, Ramprasad Saptharishi, and Anamay Tengse. If VNP is hard, then so are equations for it. In *39th International Symposium on Theoretical Aspects of Computer Science, STACS'22*, pages 44:1–44:13, 2022. 3

[LO23] Jiatu Li and Igor C Oliveira. Unprovability of strong complexity lower bounds in bounded arithmetic. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1051–1057, 2023. 2

[LST25] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. *Journal of the ACM*, 72(4):1–35, 2025. 1, 3, 4, 5, 6, 8, 11, 24, 33, 34, 55

[LST26] Jiaqi Lu, Rahul Santhanam, and Iddo Tzameret. $\mathrm{AC}^0[p]$-frege cannot efficiently prove that constant-depth algebraic circuit lower bounds are hard. In *17th Innovations in Theoretical Computer Science Conference, ITCS '26*, 2026. arXiv preprint arXiv:2509.16824. 8, 9, 12

[Min79] Henryk Minc. Evaluation of permanents. *Proceedings of the Edinburgh Mathematical Society*, 22(1):27–32, 1979. 5, 44

[MP20] Moritz Müller and Ján Pich. Feasibly constructive proofs of succinct weak circuit lower bounds. *Annals of Pure and Applied Logic*, 171(2):102735, 2020. 8

[MT14] Sebastian Müller and Iddo Tzameret. Short propositional refutations for dense random 3cnf formulas. *Annals of Pure and Applied Logic*, 165(12):1864–1918, 2014. 31

[Mul86] Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 338–339, 1986. 6

[Oli25] Igor C Oliveira. Sigact news complexity theory column 124 meta-mathematics of computational complexity theory. *ACM SIGACT News*, 56(1):41–68, 2025. 2

[Raz95a] Alexander A. Razborov. Bounded arithmetic and lower bounds in boolean complexity. In *Clote, P., Remmel, J., eds. Feasible Mathematics II. Progress in Computer Science and Applied Logic*, volume 13, pages 344–86. Birkhauser, 1995. 2

[Raz95b] Alexander A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of bounded arithmetic. *Izv. Ross. Akad. Nauk Ser. Mat.*, 59(1):201–224, 1995. 2

[Raz96] Alexander A. Razborov. Lower bounds for propositional proofs and independence results in bounded arithmetic. In *Automata, Languages and Programming, 23rd International Colloquium, ICALP '96*, pages 48–62. Springer, 1996. 2

[Raz98] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, 1998. 2, 6, 9, 21

[Raz04a] Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004. 2, 6, 7, 9

[Raz04b] Alexander A. Razborov. Resolution lower bounds for perfect matching principles. *Journal of Computer and System Sciences*, 2004. 6

[Raz10] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010. 12

[Raz15] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, 181:415–472, 2015. 2, 6, 15

[Raz16a] Alexander Razborov. Propositional proof complexity: Fifteen (or so) years after. Talk at "A Celebration of Mathematics and Computer Science. Celebrating Avi Wigderson's 60th Birthday October 5 - 8, 2016". https://youtu.be/7LfW6VTW8zo?t=2722, 2016. 2

[Raz16b] Alexander A. Razborov. Guest column: Proof complexity and beyond. *SIGACT News*, 47(2):66–86, 2016. 2

[Raz21] Alexander Razborov. P, NP and proof complexity. Talk at "SAT and Foundations of Mathematics", Simons Institute. https://youtu.be/xx4mxcqAl5A?t=2333, 2021. 2

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. System Sci.*, 55(1, part 1):24–35, 1997. 2, 3

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the Annual ACM Symposium on the Theory of Computing 1987*, pages 77–82, 1987. 8

[ST0] Rahul Santhanam and Iddo Tzameret. Iterated lower bound formulas: A diagonalization-based approach to proof complexity. *SIAM Journal on Computing*, 0(0):STOC21–313–STOC21–349, 0. 8, 9

[ST21] Rahul Santhanam and Iddo Tzameret. Iterated lower bound formulas: a diagonalization-based approach to proof complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, pages 234–247, New York, NY, USA, 2021. Association for Computing Machinery. 12

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. 9

[TC21] Iddo Tzameret and Stephen A Cook. Uniform, integral, and feasible proofs for the determinant identities. *Journal of the ACM (J. ACM)*, 68(2):1–80, 2021. 1, 4, 5, 6, 33

[Val79] 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. 10, 11