Short Propositional Refutations for Dense Random 3CNF Formulas[☆]

Sebastian Müller[a,1], Iddo Tzameret[b,2]

[a] *NII, National Institute of Informatics, Tokyo*
[b] *Computer Science Department, Royal Holloway, University of London*

Abstract

Random 3CNF formulas constitute an important distribution for measuring the average-case behavior of propositional proof systems. Lower bounds for random 3CNF refutations in many propositional proof systems are known. Most notable are the exponential-size resolution refutation lower bounds for random 3CNF formulas with $\Omega(n^{1.5-\varepsilon})$ clauses (Chvátal and Szemerédi [14], Ben-Sasson and Wigderson [10]). On the other hand, the only known non-trivial upper bound on the size of random 3CNF refutations in a non-abstract propositional proof system is for resolution with $\Omega(n^2/\log n)$ clauses, shown by Beame et al. [6]. In this paper we show that already standard propositional proof systems, within the hierarchy of Frege proofs, admit short refutations for random 3CNF formulas, for sufficiently large clause-to-variable ratio. Specifically, we demonstrate polynomial-size propositional refutations whose lines are $\boldsymbol{TC}^0$ formulas (i.e., $\boldsymbol{TC}^0$-Frege proofs) for random 3CNF formulas with n variables and $\Omega(n^{1.4})$ clauses.

The idea is based on demonstrating efficient propositional correctness proofs of the random 3CNF unsatisfiability witnesses given by Feige, Kim and Ofek [22]. Since the soundness of these witnesses is verified using spectral techniques, we develop an appropriate way to reason about eigenvectors in propositional systems. To carry out the full argument we work inside weak formal systems of arithmetic and use a general translation scheme to propositional proofs.

[☆] An extended abstract of this paper appeared in *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic In Computer Science* (LICS), pp. 501-510. 25-28 June 2012.

[1] Supported by the DAAD FitWeltweit.

[2] Email: `iddo.tzameret@gmail.com`. Supported in part by the National Natural Science Foundation of China Grant 61373002.

1. Introduction

This paper deals with the average complexity of propositional proofs. Our aim is to show that standard propositional proof systems, within the hierarchy of Frege proof systems, admit short random 3CNF refutations for a sufficiently large clause-to-variable ratio, and also can outperform resolution for random 3CNF formulas in this ratio. Specifically, we show that most 3CNF formulas with n variables and at least $cn^{1.4}$ clauses, for a sufficiently large constant c, have polynomial-size in n propositional refutations whose proof-lines are constant depth circuits with threshold gates (namely, $\boldsymbol{TC}^0$-Frege proofs). This is in contrast to resolution (that can be viewed as depth-1 Frege) for which it is known that most 3CNF formulas with at most $n^{1.5-\epsilon}$ clauses (for $0 < \epsilon < \frac{1}{2}$) do not admit sub-exponential refutations [14, 10].

The main technical contribution of this paper is a propositional characterization of the random 3CNF unsatisfiability witnesses given by Feige at al. [22]. In particular we show how to carry out certain spectral arguments inside weak propositional proof systems such as $\boldsymbol{TC}^0$-Frege. The latter should hopefully be useful in further propositional formalizations of spectral arguments. This also places a stream of recent results on efficient refutation algorithms using spectral arguments—beginning in the work of Goerdt and Krivelevich [26] and culminating in Feige et al. [22]—within the framework of propositional proof complexity. Loosely speaking, we show that all these refutation algorithms and witnesses, considered from the perspective of propositional proof complexity, are not stronger than $\boldsymbol{TC}^0$-Frege.

1.1. Background in proof complexity

Propositional proof complexity is the systematic study of the efficiency of proof systems establishing propositional tautologies (or dually, refuting unsatisfiable formulas). *Abstractly* one can view a propositional proof system as a deterministic polynomial-time algorithm A that receives a string π ("the proof") and a propositional formula Φ such that there exists a π with $A(\pi, \Phi) = 1$ iff Φ is a tautology. Such an A is called an *abstract proof system* or a *Cook-Reckhow proof system* due to [18]. Nevertheless, most research in proof complexity is dedicated to more concrete or structured models, in which proofs are sequences of lines, and each line is derived from previous lines by "local" and sound rules.

Perhaps the most studied family of propositional proof systems are those coming from propositional logic, known as Frege systems, and their fragments (and extensions). In this setting, proofs are written as sequences of

Boolean formulas (proof-lines) where each line is either an axiom or was derived from previous lines by means of simple sound derivation rules. The *complexity* of a proof is just the number of symbols it contains, that is, the total size of formulas in it. Different proof systems are compared via the concept of *polynomial simulation*: a proof system P polynomially-simulates another proof system Q if there is a polynomial-time computable function f that maps Q-proofs to P-proofs of the same tautologies. The definition of Frege systems is sufficiently robust, in the sense that different formalizations can polynomially simulate each other [41]. We call two systems that mutually simulate one another polynomially equivalent.

An example of a propositional proof system that is polynomially equivalent to Frege systems is Gentzen's sequent calculus LK. The basic building block of this calculus is a sequent, which is a relation $\Gamma \longrightarrow \Delta$ between two sequences of formulas Γ and Δ. The sequent's intended interpretation is that the conjunction of all formulas in Γ implies the disjunction of the formulas in Δ. Again, a proof of a sequent $\Gamma \longrightarrow \Delta$ is a sequence of sequents, abiding the derivation rules of LK and ending with $\Gamma \longrightarrow \Delta$. A proof of a formula φ in this system is a proof of the sequent $\longrightarrow \varphi$. As the sequent calculus is easier to analyse than arbitrary Frege systems, we will use this calculus as our starting point to define the other necessary calculi. A more detailed exposition is given in Section 2. From now on, a Frege system will mean any proof system that is polynomially equivalent to the Frege system depicted in the preceding paragraph. Especially, it will also include LK.

It is common to consider fragments (or extensions) of Frege proof systems induced by restricting the proof-lines to contain presumably weaker (or stronger) circuit classes than Boolean formulas. This stratification of Frege proof systems is thus analogous to that of Boolean circuit classes: Frege proofs consist of Boolean formulas (i.e., $\mathbf{NC}^1$) as proof-lines, $\boldsymbol{TC}^0$-Frege (also known as Threshold Logic) consists of $\boldsymbol{TC}^0$ proof-lines, Bounded Depth Frege has $\mathbf{AC}^0$ proof-lines, depth-d Frege has circuits of depth-d proof-lines, etc. In this framework, the resolution system can be viewed as *depth*-1 *Frege*. Similarly, one usually considers extensions of the Frege system such as $\mathbf{NC}^i$-Frege, for $i > 1$, and $\mathbf{P/poly}$-Frege (the latter is polynomially equivalent to the known Extended Frege system, as shown by Jeřábek [32]). Restrictions (and extensions) of Frege proof systems form a hierarchy with respect to polynomial simulations, though it is open whether the hierarchy is proper.

It thus constitutes one of the main goals of proof complexity to understand the above hierarchy of Frege systems, and to separate different propositional proof systems, that is, to show that one proof system does not polynomially simulate another proof system. These questions also relate in a certain sense to the hierarchy of Boolean circuits (from $\mathbf{AC}^0$ through $\mathbf{AC}^0[\mathrm{p}]$, $\boldsymbol{TC}^0$, $\mathbf{NC}^1$, and so forth; see [16]). Many separations between propositional proof systems (not just in the Frege hierarchy) are known. In the case of Frege proofs there are already known separations between certain fragments of it (e.g., separation of depth-d Frege from depth $d+1$ Frege was shown by Krajíček [33]). It is also known that $\boldsymbol{TC}^0$-Frege is strictly stronger than both resolution and bounded depth Frege proof systems, since, e.g., $\boldsymbol{TC}^0$-Frege admits polynomial-size proofs of the propositional pigeonhole principle, while resolution and bounded depth Frege do not (see [29] for the resolution lower bound, [1] for the bounded depth Frege lower bound and [17] for the corresponding $\boldsymbol{TC}^0$-Frege upper bound).

Average-case proof complexity via the random 3CNF model. Much like in algorithmic research, it is important to know the average-case complexity of propositional proof systems, and not just their worst-case behavior. To this end one usually considers the model of random 3CNF formulas, where m clauses with three literals each, out of all possible $2^3 \cdot \binom{n}{3}$ clauses with n variables, are chosen independently, with repetitions (however, other possible distributions have also been considered in the literature; for a short discussion on these distributions see Section 1.3). When m is greater than cn for some sufficiently large c (say, $c = 5$), it is known that with high probability a random 3CNF is unsatisfiable. (As m gets larger the task of refuting the 3CNF becomes easier since we have more constraints to use.) In average-case analysis of proofs we investigate whether such unsatisfiable random 3CNFs also have short (polynomial-size) refutations in a given proof system. The importance of average-case analysis of proof systems is that it gives us a better understanding of the complexity of a system than merely the worst-case analysis. For example, if we separate two proof systems in the average case—i.e., show that for almost all 3CNFs one proof system admits polynomial-size refutations, while the other system does not—we establish a stronger separation.

Until now only weak proof systems like resolution and Res(k) (for $k \leq \sqrt{\log n / \log\log n}$; the latter system introduced in [35] is an extension of resolution that operates with kDNF formulas) and polynomial calculus (and an extension of it) were analyzed in the random 3CNF model; for these systems exponential lower bounds are known for random 3CNFs (with varying number of clauses) [14, 6, 10, 5, 42, 2, 9, 3, 25]. For random 3CNFs with n variables and $n^{1.5-\epsilon}$ $(0 < \epsilon < \frac{1}{2})$ clauses it is known that there are no sub-exponential size resolution refutations [10]. For many proof systems, like cutting planes (CP) and bounded depth Frege ($\mathbf{AC}^0$-Frege), it is a major open problem to prove random 3CNF lower bounds (even for number of clauses near the threshold of unsatisfiability, e.g., random 3CNFs with n variables and $5n$ clauses). The results mentioned above only concern lower bounds. On the other hand, to the best of our knowledge, the only known non-trivial polynomial-size *upper bound* on random kCNF refutations in any non-abstract propositional proof system is for resolution. This is a result of Beame et al. [6], and it applies for fairly large number of clauses (specifically, $\Omega(n^{k-1}/\log n)$).

Efficient refutation algorithms. A different kind of results on refuting random kCNFs were investigated in Goerdt and Krivelevich [26] and subsequent works by Goerdt and Lanka [27], Friedman, Goerdt and Krivelevich [24], Feige and Ofek [23] and Feige [21]. Here, one studies efficient refutation *algorithms* for kCNFs. Specifically, an *efficient refutation algorithm* receives a kCNF (above the unsatisfiability threshold) and outputs either "unsatisfiable" or "don't know"; if the algorithm answers "unsatisfiable" then the kCNF is required to be indeed unsatisfiable; also, the algorithm should output "unsatisfiable" with high probability (which by definition, is also the correct answer). Such refutation algorithms can be viewed as *abstract* proof systems (according to the definition in Subsection 1.1) having short proofs on the average-case: $A(\Phi)$ is a deterministic polytime machine whose input is only kCNFs (we can think of the proposed proof π input as being always the empty string). On input Φ the machine A runs the refutation algorithm and answers 1 iff the refutation algorithm answers "unsatisfiable"; otherwise, A can decide, e.g. by brute-force search, whether Φ is unsatisfiable or not. (In a similar manner, if the original efficient refutation algorithm is *non-deterministic* then we also get an abstract proof system for kCNFs; now the proof π that A receives is the description of an accepting run of the refutation algorithm.)

Goerdt and Krivelevich [26] initiated the use of *spectral methods* to devise efficient algorithms for refuting kCNFs. The idea is that a kCNF with n variables can be associated with a graph on n vertices (or directly with a certain matrix). It is possible to show that certain properties of the associated graph witness the unsatisfiability of the original kCNF. One then uses a spectral method to give evidence for the desired graph property, and hence to witness the unsatisfiability of the original kCNF. Now, if we consider a random kCNF then the associated graph essentially becomes random too, and so one may show that the appropriate property witnessing the unsatisfiability of the kCNF occurs with high probability in the graph. The best (with respect to number of clauses) refutation algorithms devised in this way work for 3CNFs with at least $\Omega(n^{1.5})$ clauses [23].

Continuing this line of research, Feige, Kim and Ofek [22] considered efficient *non-deterministic* refutation algorithms (in other words, efficient *witnesses* for unsatisfiability of 3CNFs). They established the currently best (with respect to the number of clauses) efficient, alas non-deterministic, refutation procedure: they showed that with probability converging to 1 a random 3CNF with n variables and at least $cn^{1.4}$ clauses has a polynomial-size witness, for sufficiently big constant c.

The result in the current paper shows that all the above refutation algorithms, viewed as abstract proof systems, *are not stronger (on average) than* $\boldsymbol{TC}^0$*-Frege.* The short $\boldsymbol{TC}^0$-Frege refutations will be based on the witnesses from [22], and so the refutations hold for the same clause-to-variable ratio as in that paper.

1.2. Our result

The main result of this paper is a polynomial-size upper bound on refutations of random 3CNF formulas in a proof system operating with constant-depth threshold circuits (known as Threshold Logic or $\boldsymbol{TC}^0$-Frege; see Definition 2.4). Since Frege and Extended Frege proof systems polynomially simulate $\boldsymbol{TC}^0$-Frege proofs, the upper bound holds for these proof systems as well. (The actual formulation of $\boldsymbol{TC}^0$-Frege is not important since different formulations, given in [13, 37, 11, 40, 17], polynomially simulate each other.)

Theorem 1. *With probability* $1-o(1)$ *a random 3CNF formula with* n *variables and* $cn^{1.4}$ *clauses (for a sufficiently large constant* c*) has polynomial-size* $\boldsymbol{TC}^0$*-Frege refutations.*

Beame, Karp, Pitassi, and Saks [6] and Ben-Sasson and Wigderson [10] showed that with probability $1 - o(1)$ resolution does not admit subexponential refutations for random 3CNF formulas when the number of clauses is at most $n^{1.5-\epsilon}$, for any constant $0 < \epsilon < 1/2$.[3] Therefore, Theorem 1 shows that $\boldsymbol{TC}^0$-Frege has an exponential speed-up over resolution for random 3CNFs with at least $cn^{1.4}$ clauses (when the number of clauses does not exceed $n^{1.5-\epsilon}$, for $0 < \epsilon < 1/2$).

[3] Beame *et al.* [6] showed such a lower bound for $n^{5/4-\epsilon}$ number of clauses (for any constant $0 < \epsilon < 1/4$). Ben-Sasson and Wigderson [10] introduced the size-width tradeoff that enabled them to prove an exponential lower bound for random 3CNF formulas with $n^{1.5-\epsilon}$ number of clauses (for any constant $0 < \epsilon < 1/2$), but the actual proof for this specific clause-number appears in [7].

Potential significance of our work and its motivations. It is well known that most contemporary SAT-solvers are based on the resolution proof system. Formally, this means that these SAT-solvers use a backtracking algorithm that branch on a single variable and construct in effect a resolution refutation (in case the CNF instance considered is unsatisfiable). (The original backtracking algorithm DPLL constructs a *tree-like* resolution refutation [20, 19].) It has been known since [14] that resolution is weak in the average case. Our work gives further impetus to the quest to build SAT-solvers based on stronger proof systems than resolution. Although there is little hope to devise polynomial-time algorithms for constructing minimal $\boldsymbol{TC}^0$-Frege proofs or even resolution refutations (this stems from the conditional non-automatizability results for $\boldsymbol{TC}^0$-Frege and resolution, proved in [11] and [4], respectively), practical experience shows that current resolution-based SAT-solvers are quite powerful. Therefore, our random 3CNF upper bounds give more theoretical justification for an attempt to extend SAT-solvers beyond resolution.

Our result also advances the understanding of the relative strength of propositional proof systems: proving non-trivial upper bounds clearly rules out corresponding lower bound attempts. We conjecture that random 3CNF upper bounds similar to Theorem 1 could be achieved even for systems weaker than $\boldsymbol{TC}^0$-Frege on the expense of at most a quasipolynomial increase in the size of proofs. This might help in understanding the limits of known techniques used to prove lower bounds for random 3CNFs on resolution and Res(k) refutations.

The main result also contributes to our understanding (and possibly to the development of) refutation algorithms, by giving an explicit logical characterization of the Feige et al. [22] witnesses. This places a stream of recent results on refutation algorithms using spectral methods, beginning in Goerdt and Krivelevich [26], in the propositional proof complexity setting (showing essentially that these algorithms can be carried out already in $\boldsymbol{TC}^0$-Frege). This is a non-trivial job, especially because of the need to propositionally simulate spectral arguments. Moreover, our formalization of the spectral argument and its short propositional proofs might help in formalizing different arguments based on spectral techniques (e.g., reasoning about expander graphs).

1.3. Relation to previous work

The proof complexity of random 3CNF formulas has already been discussed above: for weak proof systems like resolution and Res(k) there are known exponential lower bounds with varying number of clauses; with respect to upper bounds, there are known polynomial size resolution refutations of random 3CNF formulas with $\Omega(n^2/\log n)$ clauses [6]. Below we shortly discuss several known upper and lower bounds on refutations of *different* distributions than the random 3CNF model (this is not an exhaustive list of all distributions studied).

Ben-Sasson and Bilu [8] have studied the complexity of refuting random 4-Exactly-Half SAT formulas. This distribution is defined by choosing at random m clauses out of all possible clauses with 4 literals over n variables. A set of clauses is *4-exactly-half satisfiable* iff there is an assignment that satisfies exactly two literals in each clause. It is possible to show that when $m = cn$, for sufficiently large constant c, a random 4-Exactly-Half SAT formulas with m clauses and n variables is unsatisfiable with high probability. Ben-Sasson and Bilu [8] showed that almost all 4-Exactly-Half SAT formulas with $m = n \cdot \log n$ clauses and n variables do not have sub-exponential resolution refutations. On the other hand, [8] provided a polynomial-time refutation algorithm for 4-Exactly-Half SAT formulas.

Another distribution on unsatisfiable formulas that is worth mentioning is 3-LIN formulas over the two element field $\mathbb{F}_2$, or equivalently 3XOR formulas. A 3-LIN formula is a collection of linear equations over $\mathbb{F}_2$, where each equation has precisely three variables. When the number of randomly chosen linear equations with 3 variables is large enough, one obtains that with high probability the collection is unsatisfiable (over $\mathbb{F}_2$). It is possible to show that the polynomial calculus proof system (see [15] for a definition), as well as $\boldsymbol{TC}^0$-Frege, can efficiently refute such random instances with high probability, by simulating Gaussian elimination.

A different type of distribution over unsatisfiable CNF formulas can possibly be constructed from the formulas (termed *proof complexity generators*) in Krajíček [36]. We refer the reader to [36] for more details on this.

Subsequent developments. In the current work we show how to prove several combinatorial and spectral arguments with polynomial-size $\boldsymbol{TC}^0$-Frege proofs. Following our work (which appeared as [38]), [44] showed that carrying (only parts of) these combinatorial arguments in a proof system that possesses the feasible interpolation property, is sufficient to improve the state-of-the-art in refutation algorithms: namely, giving polynomial-size proofs of the so-called *3XOR-principle* (Lemmas 5.6 and 5.7) in a propositional proof system that has feasible interpolation implies a polynomial-time *deterministic* refutation algorithm for random 3CNF formulas with n variables and $\Omega(n^{1.4})$ clauses.

1.4. The structure of the argument

Here we outline informally (and in some places in a simplified manner) the structure of the proof of the main theorem. We need to construct certain $\boldsymbol{TC}^0$-Frege proofs. Constructing such propositional proofs directly is technically cumbersome, and so we opt to construct them indirectly by using a first-order (two-sorted) characterization of (short proofs in) $\boldsymbol{TC}^0$-Frege: we use the theory $\boldsymbol{VTC}^0$ introduced in [40] (we tightly follow [17]). When restricted to proving only statements of a certain form (formally, Σ_0^B formulas), the theory $\boldsymbol{VTC}^0$ characterizes (uniform) polynomial-size $\boldsymbol{TC}^0$-Frege proofs.

The construction of polynomial-size $\boldsymbol{TC}^0$-Frege refutations for random 3CNF formulas will consist of the following steps:

I. Formalize the following statement as a first-order formula:

$$\begin{aligned}\forall \text{ assignment } A \ \big(&\mathbf{C} \text{ is a 3CNF and } w \text{ is its FKO unsatisfiability witness} \longrightarrow \\ &\text{exists a clause } C_i \text{ in } \mathbf{C} \text{ such that } C_i(A) = 0\big),\end{aligned} \tag{1}$$

where an *FKO witness* is a suitable formalization of the unsatisfiability witness defined by Feige, Kim and Ofek [22]. The corresponding predicate is called *the FKO predicate.*

II. Prove formula (1) in the theory $\boldsymbol{VTC}^0$.

III. Translate the proof in Step II into a family of propositional $\boldsymbol{TC}^0$-Frege proofs (of the family of propositional translations of (1)). By Theorem 3.25 (proved in [17]), this will be a polynomial-size propositional proof (in the size of $\mathbf{C}$). The translation of (1) will consist of a family of propositional formulas of the form:

$$\begin{aligned}&[\![\mathbf{C} \text{ is a 3CNF and } w \text{ is its FKO unsatisfiabiliy witness}]\!] \longrightarrow \\ &\qquad [\![\text{exists a clause } C_i \text{ in } \mathbf{C} \text{ such that } C_i(A) = 0]\!],\end{aligned} \tag{2}$$

where $[\![\cdot]\!]$ denotes the mapping from first-order formulas to families of propositional formulas. By the nature of the propositional translation (second-sort) variables in the original first-order formula translate into a collection of propositional variables. Thus, (2) will consist of propositional variables derived from the variables in (1).

IV. For the next step we first notice the following two facts:

(i) Assume that $\underline{\mathbf{C}}$ is a random 3CNF with n variables and $cn^{1.4}$ clauses, for a sufficiently large constant c (we use underlined symbols to refer to *objects*, like a 3CNF $\underline{\mathbf{C}}$, instead of *variable-symbols in the language*, like $\mathbf{C}$). By [22], with high probability there exists an FKO unsatisfiability witness $\underline{w}$ for $\underline{\mathbf{C}}$. Both $\underline{w}$ and $\underline{\mathbf{C}}$ can be encoded as finite sets of numbers, as required by the predicate for 3CNF and the FKO predicate in (1). Let us identify $\underline{w}$ and $\underline{\mathbf{C}}$ with their encodings. Then, assuming (1) was formalized correctly, assigning $\underline{w}$ and $\underline{\mathbf{C}}$ to (1) satisfies the *premise* of the implication in (1).

(ii) Now, by the definition of the translation from first-order formulas to propositional formulas, if an object α satisfies the predicate $P(X)$ (i.e., $P(\alpha)$ is true in the standard model), then there is a propositional assignment of $0, 1$ values that satisfies the propositional translation of $P(X)$. Thus, by Item (i) above, there exists an $0, 1$ assignment ζ that satisfies the premise of (2) (i.e., the propositional translation of the premise of the implication in (1)).

In the current step we show that after assigning ζ to the conclusion of (2) (i.e., to the propositional translation of the conclusion in (1)) one obtains precisely $\neg\underline{\mathbf{C}}$ (formally, a renaming of $\neg\underline{\mathbf{C}}$, where $\neg\underline{\mathbf{C}}$ is the 3DNF obtained by negating $\underline{\mathbf{C}}$ and using the de Morgan laws).

V. Take the propositional proof obtained in (**III**), and apply the assignment ζ to it. The proof then becomes a polynomial-size $\boldsymbol{TC}^0$-Frege proof of a formula $\phi \to \neg\underline{\mathbf{C}}$, where ϕ is a propositional sentence (without variables) logically equivalent to True (because ζ satisfies it, by (**IV**)). From this, one can easily obtain a polynomial-size $\boldsymbol{TC}^0$-Frege refutation of $\underline{\mathbf{C}}$ (or equivalently, a proof of $\neg\underline{\mathbf{C}}$).

The bulk of our work lies in (**I**) and especially in (**II**). We need to formalize the necessary properties used in proving the correctness of the FKO witnesses and show that the correctness argument can be carried out in the weak theory. There are two main obstacles in this process. The first obstacle is that the correctness (soundness) of the witness is originally proved using spectral methods, which assumes that eigenvalues and eigenvectors are over the *reals*; whereas the reals are not defined in our weak theory. The second obstacle is that one needs to prove the correctness of the witness, and in particular the part related to the spectral method, *constructively* (formally in our case, inside $\boldsymbol{VTC}^0$). Specifically, linear algebra is not known to be (computationally) in $\boldsymbol{TC}^0$, and (proof-complexity-wise) it is conjectured that $\boldsymbol{TC}^0$-Frege does not admit short proofs of statements of linear algebra (more specifically still, short proofs relating to inverse matrices and the determinant properties; see [43, 31] on this).

The first obstacle is solved using rational approximations of sufficient accuracy (polynomially small errors), and showing how to carry out the proof in the theory with such approximations. The second obstacle is solved basically by constructing the argument (the main formula above) in a way that exploits non-determinism (i.e., in a way that enables supplying additional witnesses for the properties needed to prove the correctness of the original witness; e.g, all eigenvectors and all eigenvalues of the appropriate matrices in the original witness). In other words, we do not have to construct certain objects but can provide them, given the possibility to certify the property we need. Formally, this means that we put additional witnesses in the FKO predicate occurring in the main formula in (**I**) above.

1.5. Organization of the paper

The remainder of the paper is organized as follows. Section 2 contains general preliminary definitions and notations, including propositional proof systems and the $\boldsymbol{TC}^0$-Frege proof system. Section 3 contains a long exposition of the basic logical setting we use, that is, the relevant theories of (two-sorted) bounded arithmetic ($\mathbf{V}^0$ and $\boldsymbol{VTC}^0$, from [17]), and a detailed explanation of how to formalize certain proofs in these theories. This includes defining certain syntactic objects in the theories as well as counting and doing computations in the theory. Readers who already know the basics of bounded arithmetic can skip Section 3, and look only at specific parts or definitions, when needed. Section 4 provides the formalization of the main formula we prove in the theory. This formula expresses the correctness of the Feige et al. witnesses for unsatisfiability [22]. Section 5 contains the proof of the main formula, excluding the lemma establishing the spectral inequality which is deferred to a section of its own. Section 6 provides the full proof in the theory of the spectral inequality. Section 7 finally puts everything together, and shows how to obtain short propositional refutations from the proof in the theory of the main formula.

2. Preliminaries

We write $[n]$ for $\{1, \dots, n\}$. We denote by $\top, \bot$ the truth values *true* and *false*, respectively.

Definition 2.1 (3CNF)**.** *A* literal *is a propositional variable x_i or its negation $\neg x_i$. A 3-*clause *is a disjunction of three literals. A 3CNF is a conjunction of 3-clauses.*

Definition 2.2 (Random 3CNF)**.** *A* random 3CNF *is generated by choosing independently, with repetitions, m clauses with three literals each, out of all possible $2^3 \cdot \binom{n}{3}$ clauses with n variables $x_1, \ldots, x_n$.*

We say that a property holds *with high probability* when it holds with probability $1 - o(1)$.

2.1. Miscellaneous linear algebra notations

We denote by $\mathbb{R}^k$ and $\mathbb{Q}^k$ the k-dimensional real and rational vector spaces in the canonical basis $e_1, \ldots, e_k$. The vectors in these spaces are given as sequences $a = (a_1 \ldots a_k)$. In this context, for some k-dimensional vector space V and two vectors $a, b \in V$ by $\langle a, b \rangle$ we denote the *inner product* of a and b which is defined by $\langle a, b \rangle := \sum_{i=1}^{k} a_i \cdot b_i$. Two vectors a, b are *orthogonal* if $\langle a, b \rangle = 0$. The *(Euclidean) norm* of a vector a is denoted by $||a||$ and is defined as $\sqrt{\sum_{i=1}^{k} a_i^2}$. A vector a is called *normal* if $||a|| = 1$. A set of vectors is called *orthonormal* if they are pairwise orthogonal and normal. A function $f : V \longrightarrow W$ is *linear* if for all $v, w \in V$, $f(c_1 v + c_2 w) = c_1 f(v) + c_2 f(w)$. Every linear function $f : V \longrightarrow W$ can be represented by a matrix $A_f = (a_{i,j})_{i \leq dim(W), j \leq dim(V)}$. Observe that the representation depends not only on f but also on the bases of V and W. A matrix $A = (a_{i,j})$ is symmetric if $a_{i,j} = a_{j,i}$ for all i, j. If for some matrix A and vector v it holds that $Av = \lambda v$ we call v an *eigenvector* and λ an *eigenvalue* of A.

Fact 1 (cf. [30])**.** *The eigenvectors of any real symmetric matrix $A : V \longrightarrow V$ form an orthogonal basis of V, and the eigenvalues of A are all real numbers.*

2.2. Propositional proofs and $\boldsymbol{TC}^0$-Frege systems

In this section we define the notion of $\boldsymbol{TC}^0$ formulas. Then we define the propositional proof system $\boldsymbol{TC}^0$-Frege as a sequent calculus operating with $\boldsymbol{TC}^0$ formulas and prove basic properties of it. We will follow the exposition from [17]. The system we give is only one of many possibilities to define such proof systems (see e.g. [11] for a polynomially-equivalent definition).

The class of $\boldsymbol{TC}^0$ formulas consists basically of unbounded fan-in constant depth formulas with $\wedge, \vee, \neg$ and threshold gates. Formally, we define:

Definition 2.3 ($\boldsymbol{TC}^0$ formula)**.** *A $\boldsymbol{TC}^0$ formula is built from*

(i) *propositional constants $\bot$ and $\top$;*

(ii) *propositional variables p_i for $i \in \mathbb{N}$;*

(iii) *connectives $\neg$ and Th_i, for $i \in \mathbb{N}$;*

(iv) *parentheses (,) .*

Items (*i*) *and* (*ii*) *constitute the* atomic formulas*. $\boldsymbol{TC}^0$ formulas are defined inductively from atomic formulas via the connectives:*

(a) *if A is a formula, then so is $\neg A$, and*

(b) *for $n > 1$ and $i \in \mathbb{N}$, if $A_1, \ldots, A_n$ are formulas, then so is $\mathsf{Th}_i(A_1, \ldots, A_n)$.*

The depth *of a formula is the maximal nesting of connectives in it and the* size *of the formula is the total number of connectives in it.*

The semantics of the *threshold connectives* Th_i are as follows. $\mathsf{Th}_i(A_1, \ldots, A_n)$ is true if and only if at least i of the A_k are true. Therefore we will abbreviate $\mathsf{Th}_i(A_1, \ldots, A_i)$ as $\bigwedge_{k \le i} A_k$ and $\mathsf{Th}_1(A_1, \ldots, A_i)$ as $\bigvee_{k \le i} A_k$. Moreover we let $\mathsf{Th}_0(A_1, \ldots, A_n) = \top$ and $\mathsf{Th}_i(A_1, \ldots, A_n) = \bot$, for $i > n$.

The following is the sequent calculus $\boldsymbol{TC}^0$-Frege.

Definition 2.4 ($\boldsymbol{TC}^0$-Frege)**.** *A $\boldsymbol{TC}^0$-Frege proof system is a sequent calculus with the axioms*

$$A \longrightarrow A, \qquad \bot \longrightarrow, \qquad \longrightarrow \top,$$

where A is any $\boldsymbol{TC}^0$ formula, and the following derivation rules:

Weaken-left: *From the sequent $\Gamma \longrightarrow \Delta$ we may infer the sequent $\Gamma, A \longrightarrow \Delta$.*

Weaken-right: *From the sequent $\Gamma \longrightarrow \Delta$ we may infer the sequent $\Gamma \longrightarrow A, \Delta$.*

Exchange-left: *From the sequent $\Gamma_1, A_1, A_2, \Gamma_2 \longrightarrow \Delta$ we may infer the sequent $\Gamma_1, A_2, A_1, \Gamma_2 \longrightarrow \Delta$.*

Exchange-right: *From the sequent $\Gamma \longrightarrow \Delta_1, A_1, A_2, \Delta_2$ we may infer the sequent $\Gamma \longrightarrow \Delta_1, A_2, A_1, \Delta_2$.*

Contract-left: *From the sequent $\Gamma, A, A \longrightarrow \Delta$ we may infer the sequent $\Gamma, A \longrightarrow \Delta$.*

Contract-right: *From the sequent $\Gamma \longrightarrow A, A, \Delta$ we may infer the sequent $\Gamma \longrightarrow A, \Delta$.*

$\neg$-left: *From the sequent* $\Gamma \longrightarrow A, \Delta$ *we may infer the sequent* $\Gamma, \neg A \longrightarrow \Delta$.

$\neg$-right: *From the sequent* $\Gamma, A \longrightarrow \Delta$ *we may infer the sequent* $\Gamma \longrightarrow \neg A, \Delta$.

All-left: *From the sequent* $A_1, \ldots, A_n, \Gamma \longrightarrow \Delta$ *we may infer the sequent* $\mathsf{Th}_n A_1 \ldots A_n, \Gamma \longrightarrow \Delta$.

All-right: *From the sequents* $\Gamma \longrightarrow A_1, \Delta, \ldots, \Gamma \longrightarrow A_n, \Delta$ *we may infer the sequent* $\Gamma \longrightarrow \mathsf{Th}_n A_1 \ldots A_n, \Delta$.

One-left: *From the sequents* $A_1, \Gamma \longrightarrow \Delta, \ldots, A_1, \Gamma \longrightarrow \Delta$ *we may infer the sequent* $\mathsf{Th}_1 A_1 \ldots A_n, \Gamma \longrightarrow \Delta$.

One-right: *From the sequent* $\Gamma \longrightarrow A_1, \ldots, A_n, \Delta$ *we may infer the sequent* $\Gamma \longrightarrow \mathsf{Th}_1 A_1 \ldots A_n, \Delta$.

Th_i-left: *From the sequents* $\mathsf{Th}_i A_2 \ldots A_n, \Gamma \longrightarrow \Delta$ *and* $\mathsf{Th}_{i-1} A_2 \ldots A_n, A_1, \Gamma \longrightarrow \Delta$ *we may infer the sequent* $\mathsf{Th}_i A_1 \ldots A_n, \Gamma \longrightarrow \Delta$.

Th_i-right: *From the sequents* $\Gamma \longrightarrow \mathsf{Th}_i A_2 \ldots A_n, A_1, \Delta$ *and* $\Gamma \longrightarrow \mathsf{Th}_{i-1} A_2 \ldots A_n, \Delta$ *we may infer the sequent* $\Gamma \longrightarrow \mathsf{Th}_i A_1 \ldots A_n, \Delta$.

Cut: *From the sequents* $\Gamma \longrightarrow A, \Delta$ *and* $\Gamma, A \longrightarrow \Delta$ *we may infer the sequent* $\Gamma \longrightarrow \Delta$*,*

for arbitrary $\boldsymbol{TC}^0$ *formulas* A_i *and sequences* Γ, Δ *of* $\boldsymbol{TC}^0$ *formulas. The intended meaning of* $\Gamma \longrightarrow \Delta$ *is that the conjunction of the formulas in* Γ *implies the disjunction of the formulas in* Δ*. A* $\boldsymbol{TC}^0$*-frege proof of a formula* φ *is a sequence of sequents* $\pi = (S_1, \ldots, S_k)$ *such that* $S_k = \longrightarrow \varphi$ *and every sequent in it is either an axiom or was derived from previous lines by a derivation rule. The* size *of the proof* π *is the total size of all formulas in its sequents. The* depth *of the proof* π *is the maximal depth of a formula in its sequents. A* $\boldsymbol{TC}^0$*-Frege proof of* a family of formulas $\{\varphi_i : i \in \mathbb{N}\}$ *is a family of sequences* $\{(S^i_1, \ldots, S^i_{k^i}) : i \in \mathbb{N}\}$*, where each* S^i_j *is a* $\boldsymbol{TC}^0$ *formula that can be derived from some* S^i_k *for* $k < j$ *using the above rules, such that* $S^i_{k^i} = \quad \longrightarrow \varphi_i$*, and there is a* common constant c bounding the depth of every formula in all the sequences.

Proposition 2.5 (cf. [17])**.** *The proof system* $\boldsymbol{TC}^0$*-Frege is sound and complete. That is, every formula A proven in the above way is a tautology and every tautology can be derived by proofs in the above sense.*

Definition 2.6 (Polynomial simulation; separation)**.** *Let* P, Q *be two propositional proof systems that establish Boolean tautologies (or refute unsatisfiable Boolean formulas, or refute unsatisfiable CNF formulas). We say that*

P polynomially simulates Q *if there is a polynomial-time computable function f that given a Q-proof of τ outputs a P-proof of τ. If P does not polynomially simulate Q or vice versa we say that P is* separated *from Q.*

(Sometimes it is enough to talk about *weak* polynomial simulations: we say that a proof system P *weakly polynomially simulates* the proof system Q if there is a polynomial p such that for every propositional tautology τ, if the minimal Q-proof of τ is of size s then the minimal P-proof of τ is of size at most $p(s)$. We also say that P is *separated* from Q when Q does not polynomially simulate Q; but in most cases it also holds that Q does not weakly polynomially simulate P.)

For a possibly partial $\{0,1\}$ assignment $\vec{a}$ to the propositional variables, we write $\varphi[\vec{a}]$ to denote the formula φ in which propositional variables are substituted by their values in $\vec{a}$. For a proof $\pi = (\varphi_1, \ldots, \varphi_\ell)$ we write $\pi[\vec{a}]$ to denote $\pi = (\varphi_1[\vec{a}], \ldots, \varphi_\ell[\vec{a}])$. The system $\boldsymbol{TC}^0$-Frege can efficiently evaluate assignments to some of the variables of formulas in the following sense.

Claim 2.7. *Let $\varphi(\vec{p}, \vec{q})$ be a propositional formula in variables $p_1 \ldots p_{m_1}$ and $q_1 \ldots q_{m_2}$ and let $\vec{a} \in \{0,1\}^{m_1}$. If $\boldsymbol{TC}^0$-Frege proves $\varphi(\vec{p}, \vec{q})$ with a proof π_φ of length n, then it also proves $\varphi(\vec{a}, \vec{q})$ in a proof $\pi_{\varphi[\vec{a}]}$ of length n. Additionally, for any formula $\varphi(\vec{p})$ in variables $p_1 \ldots p_{m_1}$ and an assignment $\vec{a} \in \{0,1\}^{m_1}$, $\boldsymbol{TC}^0$-Frege has polynomial size proofs of either $\varphi[\vec{a}]$ or $\neg\varphi[\vec{a}]$.*

Proof sketch: Consider π_φ and substitute each occurrence of p_i by a_i. The resulting proof remains correct and proves $\varphi(\vec{a}, \vec{q})$, because every $\boldsymbol{TC}^0$-Frege rule application is still correct after the assignment.

The second claim is proved by induction on the size of φ. If $\varphi[\vec{a}]$ is true we can construct a proof by proving the (substitution instances of the) atomic formulas and then proceeding using the appropriate rules of the calculus by the way the formula is built up.

If $\varphi[\vec{a}]$ is false, then we proceed in the same way as above with $\neg\varphi[\vec{a}]$ instead of $\varphi[\vec{a}]$. $\blacksquare_{\text{Claim}}$

3. Theories of bounded arithmetic

In this section we give some of the necessary background from logic. Specifically, we present the theory $\mathbf{V}^0$ and its extension $\boldsymbol{VTC}^0$, as developed by Cook and Nguyen [17] (see also [45]). These are weak systems of arithmetic, namely, fragments of Peano Arithmetic, usually referred to as theories of Bounded Arithmetic (for other treatments of theories of bounded arithmetic see also [12, 28, 34]). The theories are (first-order) two-sorted theories, having a first sort for natural numbers and a second sort for finite sets of numbers (representing bit-strings via their characteristic functions). The theory $\mathbf{V}^0$ corresponds to bounded depth Frege, and $\boldsymbol{VTC}^0$ corresponds to $\boldsymbol{TC}^0$-Frege (see Section 3.2.5). The complexity classes $\mathbf{AC}^0$, $\boldsymbol{TC}^0$, and their corresponding function classes $\mathbf{FAC}^0$ and $\mathbf{FTC}^0$ are also defined using the two-sorted universe (specifically, elements of the first sort [numbers] are given to the machines in unary representation and elements of the second sort as binary strings).

Definition 3.1 (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:*

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

We describe the intended meaning of the symbols by considering the standard model $\mathbb{N}_2$ of two-sorted Peano Arithmetic. It consists of a first-sort universe $U_1 = \mathbb{N}$ and a second-sort universe U_2 of all finite subsets of $\mathbb{N}$. We will refer to elements of the first sort as numbers, to elements of the second sort as finite sets or strings. The constants 0 and 1 are interpreted in $\mathbb{N}_2$ as the appropriate natural numbers zero and one, respectively. The functions $+$ and $\cdot$ are the usual addition and multiplication on the universe of natural numbers, respectively. The relation $\leq$ is the appropriate "less or equal than" relation on the first-sort universe. The function $|\cdot|$ maps a finite set of numbers to its largest element plus one. The relation $=_1$ is interpreted as equality between numbers, $=_2$ is interpreted as equality between finite sets of numbers. The relation $n \in N$ holds for a number n and a finite set of numbers N if and only if n is an element of N.

We denote the first-sort (number) variables by lower-case letters $x, y, z, ...$, and the second-sort (string) variables by capital letters $X, Y, Z, ...$ Terms are built up from the constants variables and functions at our disposal as usual. We call a term that contains no string variables as a *n*umber term. We build formulas in the usual way, using two sorts of quantifiers: number quantifiers and string quantifiers. A number quantifier is said to be *bounded* if it is of the form $\exists x(x \leq t \wedge \dots)$ or $\forall x(x \leq t \rightarrow \dots)$, respectively, for some number term t that does not contain x. We abbreviate $\exists x(x \leq t \wedge \dots)$ and $\forall x(x \leq t \rightarrow \dots)$ 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 $\exists X(|X| \leq t \wedge \dots)$ or $\forall X(|X| \leq t \rightarrow \dots)$ for some number term t that does not contain X.

We abbreviate $\exists X(|X| \leq t \wedge \dots)$ and $\forall X(|X| \leq t \rightarrow \dots)$ by $\exists X \leq t$ and $\forall X \leq t$, respectively. A formula is in 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 $\exists X_1 \leq t_1 \dots \exists X_m \leq t_m \psi$ or $\forall X_1 \leq t_1 \dots \forall X_m \leq t_m \psi$, where $\psi \in \Pi_i^B$ and $\psi \in \Sigma_i^B$, respectively, and t_i does not contain X_i, for all $i = 1, \dots, m$. We write $\forall\Sigma_0^B$ to denote the universal closure of Σ_0^B. (i.e., the class of Σ_0^B-formulas that possibly have (not necessarily bounded) universal quantifiers at their front). We usually abbreviate $t \in T$, for a number term t and a string term T, as $T(t)$.

For a language $\mathcal{L} \supseteq \mathcal{L}_A^2$ we write $\Sigma_0^B(\mathcal{L})$ to denote Σ_0^B formulas in the language $\mathcal{L}$.

As mentioned before a finite set of natural numbers N represents a finite string $S_N = S_N^0 \dots S_N^{|N|-1}$ such that $S_N^i = 1$ if and only if $i \in N$. We will abuse notation and identify N and S_N.

In the context of a proof in the theory, we write n^c to mean the term $\underbrace{n \cdots n}_{c \text{ times}}$ for constant c.

The (first-order) two-sorted proof system $\mathbf{LK}^2$. For proving statements in two-sorted theories we need to specify a proof system to work with (this should not be confused with the propositional proof system for Threshold Logic (i.e., $\boldsymbol{TC}^0$-Frege) we use). We shall work with a standard (two-sorted) sequent calculus $\mathbf{LK}^2$ as defined in [17], section IV.4. This sequent calculus includes the standard logical rules of the sequent calculus for first-order logic $\mathbf{LK}$ augmented with four rules for introducing second-sort quantifiers. We also have the standard equality axioms (for first- and second-sorts) for the underlying language $\mathcal{L}_A^2$ (and when we extend the language, we assume we also add the equality axioms for the additional function and relation symbols). It is not essential to know precisely the system $\mathbf{LK}^2$ since we shall not be completely formal when proving statements in two-sorted theories.

3.1. The theory $\mathbf{V}^0$

The base theory we shall work with is $\mathbf{V}^0$ and it consists of the following axioms:

Basic 1. $x + 1 \neq 0$

Basic 2. $x + 1 = y + 1 \rightarrow x = y$

Basic 3. $x + 0 = x$

Basic 4. $x + (y + 1) = (x + y) + 1$

Basic 5. $x \cdot 0 = 0$

Basic 6. $x \cdot (y + 1) = (x \cdot y) + x$

Basic 7. $(x \leq y \wedge y \leq x) \rightarrow x = y$

Basic 8. $x \leq x + y$

Basic 9. $0 \leq x$

Basic 10. $x \leq y \vee y \leq x$

Basic 11. $x \leq y \leftrightarrow x < y + 1$

Basic 12. $x \neq 0 \rightarrow \exists y \leq x(y + 1 = x)$

L1. $X(y) \rightarrow y < |X|$

L2. $y + 1 = |X| \rightarrow X(y)$

SE. $(|X| = |Y| \wedge \forall i \leq |X| \, (X(i) \leftrightarrow Y(i))) \rightarrow X = Y$

Σ_0^B**-COMP.** $\exists X \leq y \forall z < y(X(z) \leftrightarrow \varphi(z))$, for all $\varphi \in \Sigma_0^B$
where X does not occur free in φ.

Here, the axioms **Basic 1** through **Basic 12** are the usual axioms used to define Peano Arithmetic without induction (PA^-), which settle the basic properties of addition, multiplication, ordering, and of the constants 0 and 1. The Axiom **L1** says that the length of a string coding a finite set is an upper bound to the size of its elements. **L2** says that $|X|$ gives the largest element of X plus 1. **SE** is the extensionality axiom for strings which states that two strings are equal if they code the same sets. Finally, Σ_0^B**-COMP** is the comprehension axiom scheme for Σ_0^B formulas (it is an axiom for each such formula) and implies the existence of all sets which contain exactly the elements that fulfill any given Σ_0^B property.

When speaking about theories we will always assume that the theories are two-sorted theories.

Proposition 3.2 (Corollary V.1.8. [17])**.** *The theory* $\mathbf{V}^0$ *proves the (number) induction axiom scheme for* Σ_0^B *formulas* Φ*:*

$$(\Phi(0) \wedge \forall x \, (\Phi(x) \rightarrow \Phi(x + 1))) \rightarrow \forall z \, \Phi(z).$$

In the above induction axiom, x is a number variable and Φ can have additional free variables of both sorts.

We write $\exists! y \Phi$ to denote $\exists x(\Phi(x) \wedge \forall y(\Phi(y) \rightarrow x = y))$, where y is a new variable not appearing in Φ. Also, recall that we use lower-case letters such as x (and vectors $\vec{x}$) for *number variables* (and vectors of number variables, resp.); and we use upper-case letters like X for *string variables* (and similarly for their vectors).

The following is a basic notion needed to extend our language with new function symbols:

Definition 3.3 (Two-sorted definability)**.** *Let $\mathcal{T}$ be a theory over the 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 a theory $\mathcal{T}$ iff there is a formula $\varphi(\vec{x}, y, \vec{X})$ in Φ such that $\mathcal{T}$ proves*

$$\forall \vec{x} \forall \vec{X} \exists! y \varphi(\vec{x}, y, \vec{X})$$

and it holds that[4]

$$y = f(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, y, \vec{X}). \tag{3}$$

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

$$\forall \vec{x} \forall \vec{X} \exists! Y \varphi(\vec{x}, \vec{X}, Y)$$

and it holds that

$$Y = F(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, \vec{X}, Y). \tag{4}$$

Finally, a relation $R(\vec{x}, \vec{X})$ is Φ-definable *in a theory $\mathcal{T}$ iff there is a formula $\varphi(\vec{x}, \vec{X}, Y)$ in Φ such that it holds that*

$$R(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, \vec{X}). \tag{5}$$

The formulas (3)*,* (4)*, and* (6) *are the* defining axioms *for f, F, and R, respectively.*

Definition 3.4 (Conservative extension of a theory)**.** *Let $\mathcal{T}$ be a theory in the language $\mathcal{L}$. We say that a theory $\mathcal{T}' \supseteq \mathcal{T}$ in the 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 can expand the language $\mathcal{L}$ and a theory $\mathcal{T}$ over the language $\mathcal{L}$ by adding symbols for arbitrary functions f (or relations R) to $\mathcal{L}$ and their defining axioms A_f (or A_R) to the theory $\mathcal{T}$. If the appropriate functions are definable in $\mathcal{T}$ (according to Definition 3.3) then the theory $\mathcal{T} + A_f$ $(+A_R)$ is conservative over $\mathcal{T}$. This enables one to add new function and relation symbols to the language while proving statement inside a theory; as long as these function and relation symbols are definable in the theory, every statement in the original language proved in the extended theory (with the additional defining axioms for the functions and relations) is provable

[4] Meaning it holds in the standard two-sorted model $\mathbb{N}_2$.

in the original theory over the original language. *However*, extending the language and the theory in such a way *does not guarantee* that one can use the new function symbols in the comprehension (and induction) axiom schemes. In other words, using the comprehension (and induction) axioms over the expanded language might not result in a conservative extension. Therefore, definability will not be enough for our purposes. We will show precisely in the sequel (Sections 3.1.2 and 3.2) how to make sure that a function is both definable in the theories we work with and also can be used in the corresponding comprehension and induction axiom schemes (while preserving conservativity).

When expanding the language with new function symbols we can assume that in *bounded formulas* the bounding terms possibly use function symbols from the the expanded language.[5]

3.1.1. Extending $\mathbf{V}^0$ *with new function and relation symbols*

Here we describe a process (presented in Section V.4. in [17]) by which we can extend the language $\mathcal{L}_A^2$ of $\mathbf{V}^0$ by new function symbols, obtaining a conservative extension of $\mathbf{V}^0$ that can also prove the comprehension and induction axiom schemes in the extended language.

First note that every relation or function symbol has an intended or standard interpretation over the standard model $\mathbb{N}_2$ (for instance, the standard interpretation of the binary function "+" is that of the addition of two natural numbers). If not explicitly defined otherwise, we will always assume that a defining axiom of a symbol in the language defines a symbol in a way that its interpretation in $\mathbb{N}_2$ is the standard one. Note also that we shall use the same symbol $F(\vec{x}, \vec{X})$ to denote a function and the function *symbol* in the (extended) language in the theory.

Definition 3.5 (Relation representable in a language)**.** *Let* Φ *be a set of formulas in a language* $\mathcal{L}$ *extending* $\mathcal{L}_A^2$*. We say a relation* $R(\vec{x}, \vec{X})$ *is* representable *by a formula from* Φ *iff there is a formula* $\varphi(\vec{x}, \vec{X}, Y)$ *in* Φ *such that in the standard two-sorted model* $\mathbb{N}_2$ *(and when all relation and function symbols in* $\mathcal{L}$ *get their intended interpretation), it holds that:*

$$R(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, \vec{X}). \tag{6}$$

We say that a number function $f(\vec{x}, \vec{X})$ is *polynomially bounded* if

[5] Because any definable function in a bounded theory can be bounded by a term in the original language $\mathcal{L}_A^2$ (cf. [17]).

$f(\vec{x}, \vec{X}) \leq \text{poly}(\vec{x}, |\vec{X}|)$. We say that a string function $F(\vec{x}, \vec{X})$ is *polynomially bounded* if $|F(\vec{x}, \vec{X})| \leq \text{poly}(\vec{x}, |\vec{X}|)$.

Definition 3.6 (Bit-definition)**.** *Let $F(\vec{x}, \vec{X})$ be a string function. We define the* bit-graph of F *to be the relation $R(i, \vec{x}, \vec{X})$, where i is a number variable, such that*

$$F(\vec{x}, \vec{X})(i) \leftrightarrow i < t(\vec{x}, \vec{X}) \wedge R(i, \vec{x}, \vec{X}),$$

for some number term $t(\vec{x}, \vec{X})$.

Definition 3.7 (Σ_0^B-definability from a language; Definition V.4.1.2. in [17])**.** *We say that a number function f is Σ_0^B-definable from a language $\mathcal{L} \supseteq \mathcal{L}_A^2$, if f is polynomially bounded and its graph is represented by a $\Sigma_0^B(\mathcal{L})$ formula φ. We call the formula φ the* defining axiom of f*. We say that a string function F is Σ_0^B-definable from a language $\mathcal{L} \supseteq \mathcal{L}_A^2$, if F is polynomially bounded and its bit-graph is representable by a $\Sigma_0^B(\mathcal{L})$ formula φ. We call the formula φ the* defining axiom of F *or the* bit-defining axiom of F*.*

Note: We used the term *defining axiom of a function f* in both the case where f is defined *from a language* (Definition 3.7) and in case f is definable *in the theory* (Definition 3.3). We will show in the sequel that for our purposes these two notions coincide: when we define a function from a language the function will be definable also in the relevant theory, and so the defining axiom of f from the language will be the defining axiom of f in the theory (when the theory is possibly extended conservatively to include new function symbols).

Also, note that if the graph of a function F is representable by a $\Sigma_0^B(\mathcal{L})$ formula then clearly also the bit-graph of F is representable by a $\Sigma_0^B(\mathcal{L})$ formula. Therefore, *it suffices to show a $\Sigma_0^B(\mathcal{L})$ formula representing the graph of a function F to establish that F is Σ_0^B-definable from $\mathcal{L}$.*

Definition 3.8 ($\mathbf{AC}^0$-reduction)**.** *A number function f is* $\mathbf{AC}^0$-reducible *to $\mathcal{L} \supseteq \mathcal{L}_A^2$ iff there is a possibly empty sequence of functions $F_1, \ldots, F_k$ such that F_i is Σ_0^B-definable from $\mathcal{L} \cup \{F_1, \ldots, F_{i-1}\}$, for any $i = 1, \ldots, k$, and f is Σ_0^B-definable from $\mathcal{L} \cup \{F_1, \ldots, F_k\}$.*

We now describe the standard process enabling one to extend a theory $\mathcal{T} \supseteq \mathbf{V}^0$ over the language $\mathcal{L}_A^2$ with new function symbols obtaining a conservative extension of $\mathcal{T}$ such that the new function symbols can also be used in comprehension and induction axiom schemes in the theory (see Section V.4. in [17] for the proofs):

(i) If the number function f is Σ_0^B-definable from $\mathcal{L}_A^2$, then $\mathcal{T}$ over the language $\mathcal{L}_A^2 \cup \{f\}$, augmented with the defining axiom of f, is a conservative extension of $\mathcal{T}$ and we can also prove the comprehension and induction axioms for $\Sigma_0^B(f)$ formulas.

(ii) If the string function F is Σ_0^B-definable from $\mathcal{L}_A^2$, then $\mathcal{T}$ over the language $\mathcal{L}_A^2 \cup \{F\}$, augmented with the bit-defining axiom of F, is a conservative extension of $\mathcal{T}$ and we can also prove the comprehension and induction axioms for $\Sigma_0^B(F)$ formulas.

(iii) We can now iterate the above process of extending the language $\mathcal{L}_A^2(f)$ (or equivalently, $\mathcal{L}_A^2(F)$) to conservatively add more functions $f_2, f_3, \ldots$ to the language, which can also be used in comprehension and induction axioms.

By the aforementioned and by Definition 3.8, we can extend the language of a theory with a new function symbol f, *whenever f is* $\mathbf{AC}^0$*-reducible to* $\mathcal{L}_A^2$. This results in an extended theory (in an extended language) which is conservative, and can prove the comprehension and induction axioms for formulas in the extended language. In the sequel, when defining a new function in $\mathbf{V}^0$ we may simply say that it is Σ_0^B*-definable (or bit-definable) in* $\mathbf{V}^0$ and give its Σ_0^B-defining (bit-defining, respectively) axiom (that can possibly use also previously Σ_0^B-defined (or bit defined) function symbols).

Extending the language of $\mathbf{V}^0$ with new *relation* symbols is simple: every relation $R(\vec{x}, \vec{X})$ which is representable by a $\Sigma_0^B(\mathcal{L})$ formula, where $\mathcal{L}$ is an extension of the language with new function symbols obtained as shown above, can be added itself to the language. This results in a conservative extension of $\mathbf{V}^0$ that also proves the Σ_0^B induction and comprehension axioms in the extended language.

Definition 3.9 ($\mathbf{FAC}^0$)**.** *A string (number) function is in* $\mathbf{FAC}^0$ *if it is polynomially bounded and its bit-graph (graph, respectively) is definable by a* Σ_0^B *formula in the language* $\mathcal{L}_A^2$.

3.1.2. Basic formalizations in $\mathbf{V}^0$

In this section we show how to formalize basic notions in the theory $\mathbf{V}^0$.

Characteristic function of a relation. For a given predicate R we denote by χ_R the *characteristic function* of R. If R is Σ_0^B-definable in $\mathbf{V}^0$ then χ_R is Σ_0^B-definable in $\mathbf{V}^0$, using the following defining axiom:

$$y = \chi_R(\vec{x}, \vec{X}) \leftrightarrow \Big(R(\vec{x}, \vec{X}) \to y = 1 \land \neg R(\vec{x}, \vec{X}) \to y = 0 \Big) .$$

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*, and let *left*(z), *right*(z) be the (easily Σ_0^B-definable in $\mathbf{V}^0$) projection functions of the first and second element in the pair z, respectively. It should be clear from the context when we mean $\langle a, b\rangle$ as an inner product of two vectors and when we mean it as the pairing function. We also Σ_0^B-define inductively $\langle v_1, \ldots, v_k\rangle := \langle\langle v_1, \ldots, v_{k-1}\rangle, v_k\rangle$, for any constant k. Then $\mathbf{V}^0$ proves the injectivity of the pairing function and lets us handle such pairs in a standard way.

Notation: Given a number x, coding a sequence of natural numbers of constant length k, we write $\langle x\rangle_i^k$, for $i = 1, \ldots, k$, to denote the number in the ith position in x. This is a Σ_0^B-definable function in $\mathbf{V}^0$ (defined via *left*(x), *right*(x) functions).

Working with rational numbers in the theory. Given the natural numbers, we can define the *integers* in $\mathbf{V}^0$ by identifying an integer number with a pair $\langle a, b\rangle$, such that a is its "positive" part and b is its "negative" part. We can define addition, product and subtraction of integers. All with Σ_0^B definitions.

Having the integer numbers, we define the *rational* numbers as follows: for two integer numbers a and b, the rational number a/b, is defined by the pair $\langle a, b\rangle$. We can define addition, subtraction and multiplication of rational numbers, as well as the ceiling function $\lceil\cdot\rceil$ in $\mathbf{V}^0$ by Σ_0^B definitions (see [39] for the definitions of such functions).

We need to make sure that all the computations with rational numbers are definable in the theory; namely, that all the functions used throughout the computations in the theory are Σ_1^B-definable in $\boldsymbol{VTC}^0$. In particular, we need to define the *iterated addition of rational numbers* (that is, big sums of rational numbers). However, the denominator of an iterated addition of rational numbers may not be polynomially bounded. To solve this problem we use the following strategy:

Convention: ***when using big sums of rational numbers (i.e., summation of non-constant number of rational numbers), we make sure that*** $\underline{\textit{\textbf{every rational number in the sum has a fixed denominator }} n^{2\kappa}}$***, for a constant $\kappa = 7$ and n a number variable. In other words, the denominator is fixed to be the number term n^{14}.***

For the convention to be applicable, we need to make sure the following:

Fact 2. *All big sums of rational numbers that are used in the proofs have* denominators that are divisors of $n^{2\kappa}$.

Proof of fact: By inspection of the proofs in the theory. We first observe that every computation done in the proof, with numbers taken from matrices and vectors over $\mathbb{Q}$, *starts* with rational numbers that have polynomially bounded numerators and a fixed denominator n^7, where n is a number variable.

We further inspect that every computation done in the proof (with numbers taken from matrices and vectors over $\mathbb{Q}$) will *result* in rational numbers whose numerator is polynomially bounded and whose denominator is either n^7 or n^{14} (again, for n a number variable). This stems from the following: all rational numbers are either added together or we multiply one rational variable with another one (and where the result of this multiplication will *not* be multiplied further by rational numbers). ■

Comment. *Note that rational numbers that are not taken from entries in matrices and vectors (over $\mathbb{Q}$) may have arbitrary (but still polynomially bounded) denominators and numerators.*

Under the assumption above we can easily compute big sums of rational numbers: we can first make sure that all the summands have the same denominator n^{14}; and then add the integer numerators together (where iterated addition of integers is easily Σ_1^B definable in $\boldsymbol{VTC}^0$; see Section 3.2.2) to get the numerator of the big sum.

Notation: For the sake of readability we sometimes treat an integer number m in the theory as its corresponding rational number $m/1$, thus enabling one to compute with both types. (This is easy to achieve formally. E.g., one can define a function $numones'(X)$ that outputs the corresponding rational number of the integer $numones(X)$.)

Absolute numbers. We can Σ_0^B-define in $\mathbf{V}^0$ the absolute value function for integer numbers $abs_{\mathbb{Z}}(\cdot)$ from the language $\mathcal{L}_A^2$ as follows (the function max is easily Σ_0^B-definable):

$$y = abs_{\mathbb{Z}}(x) \leftrightarrow y = \langle max(left(x) - right(x), right(x) - left(x)), 0\rangle.$$

Similarly we can Σ_0^B-define the absolute value function for rational numbers $abs_{\mathbb{Q}}(\cdot)$ in $\mathbf{V}^0$.

For simplicity, we shall suppress the subscript $\mathbb{Z}, \mathbb{Q}$ in $abs_{\mathbb{Z}}, abs_{\mathbb{Q}}$; the choice of function can be determined from the context.

Number sequences of polynomial length (over $\mathbb{N}, \mathbb{Z}$ and $\mathbb{Q}$). If we wish to talk about sequences of numbers (whether natural, integers or rationals) where the lengths of the sequences are non-constant, we have to use string variables. Using the number tupling function we can encode sequences as sets of numbers (recall that a string is identified with the finite set of numbers encoding it). Essentially, 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_0^B-defining formula for the function $seq(x, Z)$:

$$\begin{aligned} y = seq(x,Z) \leftrightarrow \big(y < |Z| \wedge Z(\langle x,y\rangle) \wedge \forall z < y \neg Z(\langle x,z\rangle)\big) \\ \vee\, (\forall z < |Z| \neg Z(\langle x,z\rangle) \wedge y = |Z|). \end{aligned} \tag{7}$$

Formula (7) 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. We write

$$Z[x]$$

to abbreviate $seq(x, Z)$.

According to the definition of the function $seq(x, Z)$ above, there might be more than one string Z that encodes the same sequence of numbers. However, we sometimes need to determine a *unique* string encoding a sequence. To this end we use a Σ_0^B formula, denoted $SEQ(y, Z)$, which asserts that Z is the lexicographically smallest string that encodes a sequence of y numbers (i.e., no string with smaller binary code encodes the same sequence). Specifically, the formula states that if $w = \langle i, j\rangle$ is in Z then j is indeed the ith element in the sequence coded by Z, and for all $y \geq j$ the pair $\langle i, y\rangle$ is not contained in Z:

$$\begin{aligned} SEQ(y,Z) \equiv \forall w < |Z|\,(Z(w) \leftrightarrow \exists i < |Z| \exists j < |Z|\,(w = \langle i,j\rangle \wedge j = Z[i])) \\ \wedge\, \forall i < y \exists j < |Z| (j = Z[i]). \end{aligned} \tag{8}$$

We define the number function $length(Z)$ to be the length of the sequence Z, as follows:

$$\ell = length(Z) \leftrightarrow SEQ(\ell, Z) \wedge \exists w < |Z| \exists j < |Z| (Z(w) \wedge w = \langle \ell - 1, j\rangle)\,.$$

The defining axiom of $length(Z)$ states that Z encodes a sequence and is the lexicographically smallest string that encodes this sequence and that the largest position in the sequence which is occupied is $\ell - 1$ (by definition of SEQ (8) there will be no pair $\langle a, b\rangle \in Z$ with $a > \ell - 1$).

Note that elements of sequences Z coded by strings are referred to as $Z[i]$, while elements of sequences x coded by a number are referred to as $\langle x\rangle_i^k$ (for k the length of the sequence x).

Array of strings. We want to encode a sequence of strings as an array. We use the relation $RowArray(x, Z)$ to denote the xth string in Z as follows (we follow the treatment in [17], Definition V.4.26, page 114).

Definition 3.10 (Array of strings)**.** *The function $RowArray(x, Z)$, denoted $Z^{[x]}$, is Σ_0^B-definable in $\mathbf{V}^0$ using the following bit-definition:*[6]

$$RowArray(x, Z)(i) \leftrightarrow (i < |Z| \wedge Z(\langle x, i\rangle)).$$

We will abuse notation and write $length(Z)$ for the length of the array Z (i.e., number of strings in Z) even when Z is a $RowArray$ (and not a sequence according to the predicate SEQ).

Functions for constructing sequences.

Definition 3.11 ($Sequence_f(y, \vec{x}, \vec{X})$)**.** *Let $f(z, \vec{x}, \vec{X})$ be a Σ_0^B-definable number function in $\mathbf{V}^0$ (or a Σ_1^B-definable number function in $\boldsymbol{VTC}^0$[see section 3.2 below]), then $Sequence_f(y, \vec{x}, \vec{X})$ is the string function Σ_0^B-definable in $\mathbf{V}^0$ (or Σ_1^B-definable in $\boldsymbol{VTC}^0$, respectively) that returns the number sequence whose jth position is $f(j, \vec{x}, \vec{X})$, for $j = 0, \dots, y$.*

In other words, $Sequence_f(y, \vec{x}, \vec{X})$ returns the graph of the function $f(z, \vec{x}, \vec{X})$ up to y (that is, the sequence $\langle f(0, \vec{x}, \vec{X}), \dots, f(y, \vec{x}, \vec{X})\rangle$). The following is the Σ_0^B-definition of the $Sequence_f(y, \vec{x}, \vec{X})$:

$$Y = Sequence_f(y, \vec{x}, \vec{X}) \leftrightarrow SEQ(y, Y) \wedge \forall z \le y\, (Y[i] = f(z, \vec{x}, \vec{X})).$$

Sequences of numbers with higher dimensions. For a constant k, let S be a k-dimensional sequence of rational numbers. We encode a sequence S as a string variable Z such that the $\langle i_1, \dots, i_k\rangle$th element in S is extracted by the function seq (defined above). Specifically, we have $S[\langle i_1, \dots, i_k\rangle] = y$ iff $\langle\langle i_1, \dots, i_k\rangle, y\rangle \in Z$ and there is no $z < y$ for which $\langle\langle i_1, \dots, i_k\rangle, z\rangle \in Z$. Accordingly, we write $Z[i_1, \dots, i_k]$ to abbreviate $seq(\langle i_1, \dots, i_k\rangle, Z)$.

[6] We use the name "RowArray" (instead of the name "Row" used in [17]).

Matrices. Given a rational $n \times n$ matrix M, we define it as a two-dimensional sequence in the manner defined above; and refer to the number at row $1 \leq i \leq n$ and column $1 \leq j \leq n$ of M as $M[i,j]$. We can define the *string* function that extracts the xth row of M, and the xth column of M, respectively, with Σ_0^B formulas as follows. First define $f(M,i,x) := M[i,x]$, $g(M,i,x) := M[x,i]$, for any $i = 0,1,\dots,n$ (for $i = 0$, the value of $M[i,x]$ and $M[x,i]$ does not matter; but this value is still defined by definition of the function *seq*). Then use Definition 3.11 to define:

$$\begin{aligned} Row(i,M) &:= Sequence_f(i,n) \\ Column(i,M) &:= Sequence_g(i,n)\,. \end{aligned}$$

3.2. The theory $\boldsymbol{VTC}^0$

It is known that $\mathbf{V}^0$ is incapable of proving basic counting statements. Specifically, it is known that the function that sums a sequence of numbers (of non-constant length) is not provably total, namely, is not Σ_1^B-definable in $\mathbf{V}^0$ (see Definition 3.3.). Therefore, if a proof involves such computations we might not be able to perform it in $\mathbf{V}^0$. The theory $\boldsymbol{VTC}^0$ extends $\mathbf{V}^0$, and is meant to allow reasoning that involves counting, and specifically to sum a non-constant sequence of numbers. The theory $\boldsymbol{VTC}^0$ was introduced in [40]; we refer the reader to Section IX.3.2. [17] for a full treatment of this theory. The Σ_0^B theorems of $\boldsymbol{VTC}^0$ correspond to polynomial-size $\boldsymbol{TC}^0$-Frege propositional proofs, which will enable us to prove the main result of this paper.

Definition 3.12 (NUMONES)**.** *Let $\delta_{\mathsf{NUM}}(y,X,Z)$ be the following Σ_0^B formula:*

$$\begin{aligned} \delta_{\mathsf{NUM}}(y,X,Z) := SEQ(y,Z) \wedge Z[0] = 0 \wedge \forall u < y((X(u) \to Z[u+1] = Z[u]+1) \\ \wedge (\neg X(u) \to Z[u+1] = Z[u])). \end{aligned} \tag{9}$$

Define NUMONES *to be the following Σ_1^B formula:*

$$\mathsf{NUMONES} := \exists Z \leq 1 + \langle y,y\rangle \delta_{\mathsf{NUM}}(y,X,Z). \tag{10}$$

Informally one can think of the sequence $Z(X)$, whose existence is guaranteed by NUMONES, as a sequence counting the number of ones in a string X, that is, the uth entry in $Z(X)$ is the number of ones appearing in the string X up to the uth position.

Definition 3.13 ($\boldsymbol{VTC}^0$)**.** *The theory $\boldsymbol{VTC}^0$ is the theory containing all axioms of $\mathbf{V}^0$ and the axiom* $\mathsf{NUMONES}$*.*

Using $\mathsf{NUMONES}$ we can define the function $numones(y, X)$ that, given y and X, returns the yth entry of $Z(X)$ via the following Σ_1^B-defining axiom

$$numones(y, X) = z \leftrightarrow \exists Z \leq 1 + \langle |X| , |X| \rangle \left(\delta_{\mathsf{NUM}}(|X| , X, Z) \wedge Z[y] = z\right). \tag{11}$$

We shall use the following abbreviation:

$$numones(X) := numones(|X| - 1, X).$$

Next we show how to obtain the functions we will use in the theory $\boldsymbol{VTC}^0$ (these will include the function *numones*).

3.2.1. Extending $\boldsymbol{VTC}^0$ with new function and relation symbols

Similar to the case of $\mathbf{V}^0$, we would like to extend the language $\mathcal{L}_A^2$ of $\boldsymbol{VTC}^0$ with new function and relation symbols, to obtain a conservative extension. Moreover, we require that the new function and relation symbols could be used in induction and comprehension axioms (while preserving conservativity). We can do this, using results from Sections I.X.3.2. and I.X.3.3. in [17], as follows.

Definition 3.14 (Number summation)**.** *For any number function $f(z, \vec{x}, \vec{X})$ define the number function $\mathsf{sum}_f(y, \vec{x}, \vec{X})$ by*[7]

$$\mathsf{sum}_f(y, \vec{x}, \vec{X}) = \sum_{i=0}^{y} f(i, \vec{x}, \vec{X}).$$

Recall that by Definition 3.7, a string (number) function F is Σ_0^B-*definable from* $\mathcal{L} \supseteq \mathcal{L}_A^2$ iff there is a Σ_0^B formula over the language $\mathcal{L}$ that bit-defines (defines, respectively) the function F (when all the functions and relation symbols in $\mathcal{L}$ get their intended interpretation).

We can use the following facts to extend the language of $\boldsymbol{VTC}^0$ with new function symbols (proved in Section IX.3.2. in [17]): if f is a (number or string) function in $\mathbf{FTC}^0$ (see below), then there is a Σ_1^B formula φ that represents its graph, and the theory $\boldsymbol{VTC}^0$ extended with the defining

[7] Note that this is a definition in the meta-theory (or in other words the standard two-sorted model).

axiom for f (using φ, as in Definition 3.7) over the language $\mathcal{L} = \mathcal{L}_A^2 \cup \{f\}$ is a conservative extension of $\boldsymbol{VTC^0}$. And by Theorem IX.3.7. in Section IX.3.2. [17], $\boldsymbol{VTC^0}$ can prove the induction and comprehension axioms for any $\Sigma_0^B(\mathcal{L})$ formula.

Thus, to extend $\boldsymbol{VTC^0}$ with new *function* symbols, by the above it suffices to show how to obtain $\mathbf{FTC}^0$ functions. For this we use the following equivalent characterizations of $\mathbf{FTC}^0$ (see Sections IX.3.2. and IX.3.3. in [17]):

Proposition 3.15 (Theorem IX.3.12., Proposition IX.3.1. in [17])**.** *The following statements are equivalent:*

1. *The function f is Σ_1^B-definable in $\boldsymbol{VTC^0}$, and is applicable inside comprehension and induction axiom schemes.*
2. *The function f is in $\mathbf{FTC}^0$.*
3. *The function f is obtained from $\mathbf{FAC}^0$ by number summation and $\mathbf{AC}^0$-reductions.*
4. *There exist a natural k and functions $f_1, \dots, f_k = f$ such that for every $i = 1, \dots, k$, the function f_i is either definable by a Σ_0^B formula in the language $\mathcal{L}_A^2 \cup \{f_1, \dots, f_{i-1}\}$ or there exists $h \in \mathcal{L}_A^2 \cup \{f_1, \dots, f_{i-1}\}$ such that $f_i = \mathsf{sum}_h$.*
5. *The function f is $\mathbf{AC}^0$-reducible to $\mathcal{L}_A^2 \cup \{numones\}$.*

Therefore, to obtain new $\mathbf{FTC}^0$ functions, and hence to extend conservatively the language of $\boldsymbol{VTC^0}$ with function symbols that can also be used in comprehension and induction axioms, we can define a function with a Σ_0^B formula in a language that contains sum_f, for f in $\mathbf{FAC}^0$, and possibly contains also other symbols already definable in $\mathbf{V}^0$. Then, we can iterate this process a finite number of times, where now sum_f is defined also for f being a function defined in a previous iteration. Since a function is in $\mathbf{FTC}^0$ iff it is Σ_1^B-definable in $\boldsymbol{VTC^0}$, new functions obtained in this way are said to be Σ_1^B*-definable in* $\boldsymbol{VTC^0}$.

To extend the language of $\boldsymbol{VTC^0}$ with new *relation* symbols, we can simply add new Σ_0^B-definable relations, using possibly relation and function symbols that were already added to the language, and specifically the *numones* function. Such relations can then be used in induction and comprehension axioms, and we shall say that they are Σ_0^B*-definable relations in* $\boldsymbol{VTC^0}$.

3.2.2. Summation in $\boldsymbol{VTC^0}$

Here we show how to express and prove basic equalities and inequalities in the theory $\boldsymbol{VTC^0}$.

Summation over natural and rational number sequences. Given a sequence X of *natural* numbers, we define the function that sums the numbers in X until the yth position by $\mathsf{sum}_{seq}(y, X)$ which is equal to $\sum_{i=0}^{y} seq(i, X)$.

To sum sequences of *rational* numbers, on the other hand, we do the following. For our purposes it is sufficient to sum polynomially bounded numbers (this is in contrast to addition of numbers encoded as *strings*); recall also (Section 3.1.2) that for our purposes it will be enough to have big sums (i.e., summation of more than 2 summands) of rational numbers with the same denominator n^{14}. Thus, the big sum function is defined only for numbers with the same denominator.

Proposition 3.16. *Let X be a sequence of rational numbers with the same denominator. Then, the number function $sum_{\mathbb{Q}}(z, X)$ that outputs $\sum_{i=0}^{z} X[i]$ is Σ_1^B-definable in $\boldsymbol{VTC}^0$.*

Proof. It suffices to show that there is a Σ_0^B formula that defines the number function $sum_{\mathbb{Q}}(z, X)$ using only number summation functions and $\mathbf{FAC}^0$ functions.

The $\mathbf{AC}^0$ function $seq(i, X)$ extracts the ith element (that is, rational number) from the sequence X (see Equation (7) above). A rational number is a pair of integers, and hence is a pair of pairs. Thus, $gp(i, X) := left(left(seq(i, X)))$ extracts the positive part of the integer numerator of the ith rational number in X, and $gn(i, X) := right(left(seq(i, X)))$ extracts the negative part of the integer numerator of the ith rational number in X. Note that both $gp(i, X)$ and $gn(i, X)$ are $\mathbf{FAC}^0$ functions. Therefore, $\mathsf{sum}_{gp}(z, X)$ equals the sum of all the positive parts in X, and $\mathsf{sum}_{gn}(z, X)$ equals the function that sums of all the negative parts of the numerators in X. The term $right(seq(i, X))$ is the common denominator of all the rational numbers in X. Hence, we can define $sum_{\mathbb{Q}}(z, X)$ as follows:

$$w = sum_{\mathbb{Q}}(z, X) \leftrightarrow w = \langle \langle \mathsf{sum}_{gp}(z, X), \mathsf{sum}_{gn}(z, X) \rangle \, , right(seq(i, X)) \rangle \tag{12}$$

□

Notation: As a corollary from Proposition 3.16, we can abuse notation as follows: for $f(y, \vec{x}, \vec{X})$ a number function mapping to the rationals we write $\mathsf{sum}_f(n, \vec{x}, \vec{X})$ to denote the sum of *rationals* $\sum_{i=0}^{n} f(i, \vec{x}, \vec{X})$, for some fixed $\vec{x}, \vec{X}$ and n. Abusing notation further, we can write in a formula in the theory simply $\sum_{i=0}^{n} f(i, \vec{x}, \vec{X})$.

Expressing vectors and operations on vectors. Vectors over $\mathbb{Q}$ are defined as sequences of rational numbers (for simplicity we shall assume that the number at the 0 position of a vector is 0). Thus the elements of vectors are rational numbers $a/n^{2\kappa}$, with a fixed denominator). Given two rational vectors $\mathbf{v}, \mathbf{u}$ of size n, their inner product, denoted $\langle \mathbf{v}, \mathbf{u} \rangle$, is defined as follows (we identify here $\mathbf{v}, \mathbf{u}$ with the string variables encoding $\mathbf{v}, \mathbf{u}$): let $f(y, \mathbf{v}, \mathbf{u})$ be the $\mathbf{FAC}^0$ number function defined by $f(y) := \mathbf{v}[y] \cdot \mathbf{u}[y]$. Then the inner product of $\mathbf{v}$ and $\mathbf{u}$ is defined by

$$\mathit{innerprod}(\mathbf{v}, \mathbf{u}) := \mathit{sum}_{\mathbb{Q}} \left(\mathit{length}(\mathbf{v}) + 1, \mathit{Sequence}_f(\mathit{length}(\mathbf{v}) + 1) \right).$$

The function that adds two rational vectors is easily seen to be in $\mathbf{FAC}^0$ (use Definition 3.11 to construct a sequence, where each entry in the sequence is the addition of the corresponding entries of the two vectors).

Expressing products of matrices and vectors. Let $\mathbf{v}$ be an n-dimensional rational vector and let M be an $n \times n$ rational matrix. Assume that $f(z, M, \mathbf{v}) := \mathit{innerprod}(\mathit{Row}(z, M), \mathbf{v})$. We Σ_1^B-define in $\boldsymbol{VTC^0}$ the product $M\mathbf{v}$ as follows:

$$\mathit{Matvecprod}(M, \mathbf{v}) := \mathit{Sequence}_f(\mathit{length}(\mathbf{v}) + 1, M, \mathbf{v}).$$

Notation: When reasoning in the theory $\boldsymbol{VTC^0}$ we sometimes abuse notation and write $\mathbf{v} \cdot \mathbf{u}$ or $\langle \mathbf{v}, \mathbf{u} \rangle$ instead of $\mathit{innerprod}(\mathbf{u}, \mathbf{v})$, and $M\mathbf{v}$ instead of $\mathit{Matvecprod}(M, \mathbf{v})$, and $\mathbf{u}^t M \mathbf{v}$ instead of $\langle \mathbf{u}, M\mathbf{v} \rangle$.

3.2.3. Counting in $\boldsymbol{VTC^0}$

Here we present basic statements involving counting of certain objects and sets, provable in $\boldsymbol{VTC^0}$.

Notation: When reasoning in the theory $\boldsymbol{VTC^0}$, we will say that a family of sets $B_0, \ldots, B_\ell$ which are Σ_0^B-definable in $\boldsymbol{VTC^0}$ *forms a partition of* $\bigcup_{i=0}^{\ell} B_i := \{r \ : \ \exists i \le \ell, B_i(r)\}$ whenever $\boldsymbol{VTC^0}$ proves that (i) $\bigcup_{i=0}^{\ell} B_i = B$, and (ii) $B_i \cap B_j = \emptyset$, for all $0 \le i \ne j \le \ell$.

Proposition 3.17 (Some counting in $\boldsymbol{VTC^0}$)**.** *Let $B_1, \ldots, B_\ell$ be a family of Σ_0^B-definable sets in $\boldsymbol{VTC^0}$ that partition the set B (ℓ may be a variable). Then, $\boldsymbol{VTC^0}$ proves:*

$$\mathit{numones}(B) = \sum_{i=1}^{\ell} \mathit{numones}(B_i).$$

Proof. We proceed by induction on ℓ to show that for every $0 \leq y \leq \max\{B_1, \ldots, B_\ell\}$:

$$numones(y, B_1 \cup \cdots \cup B_\ell) = \sum_{i=1}^{\ell} numones(y, B_i).$$

Base case: $\ell = 1$. Thus, $B = B_1$ and so we need to prove only $numones(y, B_1) = \sum_{i=1} numones(y, B_i)$. Since $\boldsymbol{VTC}^0$ proves that a summation that contains only one summand B_1 equals B_1 we are done.

Induction step: $\ell > 1$. We have $B = \bigcup_{i=1}^{\ell} B_i = (\bigcup_{i=1}^{\ell-1} B_i) \cup B_\ell$. Assume by way of contradiction that $(\bigcup_{i=1}^{\ell-1} B_i) \cap B_\ell \neq \emptyset$. Then $\boldsymbol{VTC}^0$ can prove that this contradicts the assumption that $B_i \cap B_j = \emptyset$, for all $i \neq j$ (which holds since the B_i's form a partition of B). Hence, $(\bigcup_{i=1}^{\ell-1} B_i) \cap B_\ell = \emptyset$, and by Claim 3.18 (proved below):

$$\begin{aligned} numones(y, B) &= numones(y, \bigcup_{i=1}^{\ell-1} B_i) + numones(y, B_\ell) \\ &= \sum_{i=1}^{\ell-1} numones(y, B_i) + numones(y, B_\ell) \quad \text{(by induction hypothesis)} \\ &= \sum_{i=1}^{\ell} numones(y, B_i). \end{aligned}$$

It remains to prove the following:

Claim 3.18. *(In $\boldsymbol{VTC}^0$) let A, B be two sets such that $A \cap B = \emptyset$, then for all $0 \leq y \leq \max\{|A|, |B|\}$:*

$$numones(y, A \cup B) = numones(y, A) + numones(y, B).$$

Proof of claim: We proceed by induction on y, using the defining axiom of *numones* (stating the existence of a counting sequence for the input string variable; see Equations (11) and (9)).

Base case: $y = 0$. The counting sequence Z for $numones(A \cup B)$ is defined such that $Z[0] = 0$. Thus,

$$0 = numones(0, A \cup B) = numones(0, A) + numones(0, B) = 0 + 0 = 0.$$

Induction step: $0 < y \leq \max\{|A|, |B|\}$. By the defining axiom of *numones* we have:

$$numones(y, A \cup B) = \begin{cases} numones(y-1, A \cup B) + 1, & y \in A \cup B; \\ numones(y-1, A \cup B), & \text{otherwise.} \end{cases} \tag{13}$$

We have to consider the following three cases:
Case 1: $y \in A$. Thus, by assumption that A and B are disjoint, we have $y \notin B$. Also, we have $y \in A \cup B$. Therefore:

$$\begin{aligned} & numones(y, A) + numones(y, B) \\ & \quad = numones(y-1, A) + 1 + numones(y, B) && \text{(since } y \in A) \\ & \quad = numones(y-1, A) + 1 + numones(y-1, B) && \text{(since } y \notin B) \\ & \quad = numones(y-1, A \cup B) + 1 && \text{(by induction hypothesis)} \\ & \quad = numones(y, A \cup B) && \text{(since } y \in A \cup B). \end{aligned}$$

Case 2: $y \in B$. This is the same as Case 1.
Case 3: $y \notin A \cup B$. This is similar to the previous cases. We omit the details. $\blacksquare_{\text{Claim}}$

$\square$

Proposition 3.19 (More counting in $\boldsymbol{VTC}^0$)**.** *Let $\varphi(x)$ be a Σ_0^B formula (possibly in an extended language of $\boldsymbol{VTC}^0$). The theory $\boldsymbol{VTC}^0$ can prove that if $Z = \{0 \leq i < m : \varphi(i)\}$ and for any $0 \leq i < m$,*

$$\gamma_i = \begin{cases} a, & \varphi(i); \\ b, & \neg\varphi(i), \end{cases}$$

then

$$\sum_{i<m} \gamma_i = a \cdot numones(Z) + b \cdot (m - numones(Z)).$$

Proof. Since $\varphi(x)$ is a Σ_0^B formula, by Section 3.2.1, we can use the comprehension axiom scheme to define, for any $0 \leq k \leq m-1$, the set:

$$Z_k := \{i \leq k : \varphi(i)\}.$$

The claim is proved by induction on k.

Base case: $k = 0$. If $\varphi(0)$ is true, then $Z_0 = \{0\}$, and so $numones(Z_0) = 1$. By assumption we have $\gamma_0 = a = a \cdot numones(Z_0) + b \cdot (1 - numones(Z_0))$.

Otherwise, $\varphi(0)$ is false and so $Z_0 = \emptyset$, implying that $numones(Z_k) = 0$. By assumption again we have $\gamma_0 = b = a \cdot numones(Z_0) + b(1 - numones(Z_0))$.

Induction step: $k > 0$.
Case 1: $\varphi(k)$ is true. Thus $Z_k(k)$ is true and also

$$numones(Z_k) = numones(Z_{k-1}) + 1, \tag{14}$$

and by assumption $\gamma_k = a$. Therefore,

$$\begin{aligned}
&\sum_{i=0}^{k} \gamma_i = \sum_{i=0}^{k-1} \gamma_i + \gamma_k = \sum_{i=0}^{k-1} \gamma_i + a \\
&= a \cdot numones(Z_{k-1}) + b \cdot (k - 1 - numones(Z_{k-1})) + a && \text{(by induction hypothesis)} \\
&= a \cdot (numones(Z_{k-1}) + 1) + b \cdot (k - 1 - numones(Z_{k-1})) && \text{(rearranging)} \\
&= a \cdot numones(Z_k) + b \cdot (k - numones(Z_k)) && \text{(by (14)).}
\end{aligned}$$

Case 2: $\varphi(k)$ is false. This is similar to Case 1. Specifically, $Z_k(k)$ is false and also

$$numones(Z_k) = numones(Z_{k-1}), \tag{15}$$

and by assumption $\gamma_k = b$. Therefore

$$\begin{aligned}
&\sum_{i=0}^{k} \gamma_i = \sum_{i=0}^{k-1} \gamma_i + \gamma_k = \sum_{i=0}^{k-1} \gamma_i + b \\
&= a \cdot numones(Z_{k-1}) + b \cdot (k - 1 - numones(Z_{k-1})) + b && \text{(by induction hypothesis)} \\
&= a \cdot numones(Z_{k-1}) + b \cdot (k - 1 - numones(Z_{k-1}) + 1) && \text{(rearranging)} \\
&= a \cdot numones(Z_k) + b \cdot (k - numones(Z_k)) && \text{(by (15)).}
\end{aligned}$$

□

For a number term t, we write $\forall x \in [t]\, \Phi$ to abbreviate $\forall x \leq t(x \geq 1 \rightarrow \Phi)$. We shall use the following proposition in Section 5 (Lemma 5.15).

Proposition 3.20. *The theory* $\boldsymbol{VTC}^0$ *proves the following statement. Let* $F(x)$ *be a string function. Let* $d < t$ *be two natural numbers and assume that any number in each set* $F(1), \ldots, F(t)$ *occurs in at most* d *many sets in* $F(1), \ldots, F(t)$*. Let* $g(x)$ *be a number function such that* $g(1), \ldots, g(t)$ *are (not necessarily distinct) numbers with* $g(i) \in F(i)$ *for all* $i \in [t]$*. Then* $numones(\{g(i) \ : \ i \in [t]\}) \geq \lceil t/d \rceil$.

Proof. Let $\mathsf{Img}(g(x)) := \{i : g(x) \in F(i)\}$ be a string function (it is Σ_0^B-definable in $\mathbf{V}^0$). By assumption

$$\forall z \in [t]\,(\mathit{numones}(\mathsf{Img}(g(z))) \leq d)\,. \tag{16}$$

Since for any $i \in [t]$, $g(i) \in F(i)$, we can prove in $\boldsymbol{VTC}^0$ that $\bigcup_{z\in[t]} \mathsf{Img}(g(z))$ equals $\{1, 2, \ldots, t\}$, and so $\boldsymbol{VTC}^0$ proves:

$$\mathit{numones}\left(\bigcup_{z\in[t]} \mathsf{Img}(g(z))\right) = t. \tag{17}$$

Claim 3.21. *(Under the assumptions of the proposition)* $\boldsymbol{VTC}^0$ *proves:*

$$\mathit{numones}\left(\bigcup_{z\in[t]} \mathsf{Img}(g(z))\right) \leq d \cdot \mathit{numones}(\{g(i)\ :\ i \in [t]\}).$$

Proof of claim: The proof follows from (16), by induction on t.
Base case: $t = 1$. We have

$$\begin{aligned}
\mathit{numones}(\cup_{z\in[t]}\mathsf{Img}(g(z))) &= \mathit{numones}(\mathsf{Img}(g(1))) \\
&\leq d \qquad\qquad \text{(by assumption)} \\
&= d \cdot \mathit{numones}(\{g(1)\}) \\
&= d \cdot \mathit{numones}(\{g(i)\ :\ i \in [t]\}).
\end{aligned}$$

Induction step:
Case 1: $g(t) \in \{g(i)\ :\ i \in [t-1]\}$. Thus,

$$\{g(i)\ :\ i \in [t{-}1]\} = \{g(i)\ :\ i \in [t]\} \qquad \text{and} \qquad \bigcup_{i\in[t-1]} \mathsf{Img}(g(i)) = \bigcup_{i\in[t]} \mathsf{Img}(g(i)). \tag{18}$$

Therefore,

$$\begin{aligned}
\mathit{numones}\left(\bigcup_{i\in[t]} \mathsf{Img}(g(i))\right) &= \mathit{numones}\left(\bigcup_{i\in[t-1]} \mathsf{Img}(g(i))\right) \\
&\leq d \cdot \mathit{numones}\left(\{g(i)\ :\ i \in [t-1]\}\right) \quad \text{(by induction hypothesis)} \\
&= d \cdot \mathit{numones}\left(\{g(i)\ :\ i \in [t]\}\right) \qquad \text{(by (18)).}
\end{aligned}$$

Case 2: $g(t) \notin \{g(i) \ : \ i \in [t-1]\}$. Thus,

$$numones(\{g(i) \ : \ i \in [t-1]\}) + 1 = numones(\{g(i) \ : \ i \in [t]\}). \qquad (19)$$

We have

$$numones \left(\bigcup_{z\in[t]} \mathsf{Img}(g(z)) \right) \leq numones \left(\bigcup_{z\in[t-1]} \mathsf{Img}(g(z)) \right) + numones\left(\mathsf{Img}(g(t))\right),$$

and by induction hypothesis

$$\begin{aligned}
&\leq d \cdot numones\left(\{g(i) \ : \ i \in [t-1]\}\right) + numones\left(\mathsf{Img}(g(t))\right) \\
&\leq d \cdot (numones(\{g(i) \ : \ i \in [t]\}) - 1) + numones\left(\mathsf{Img}(g(t))\right) \quad \text{(by (19))} \\
&\leq d \cdot (numones(\{g(i) \ : \ i \in [t]\}) - 1) + d \qquad \text{(by assumption)} \\
&= d \cdot numones(\{g(i) \ : \ i \in [t]\}).
\end{aligned}$$

■Claim

Thus, by Claim 3.21 and by (17), we get:

$$t \leq d \cdot numones(\{g(i) \ : \ i \in [t]\}),$$

which leads to $t/d \leq numones(\{g(i) \ : \ i \in [t]\})$, and since $numones(\{g(i) \ : \ i \in [t]\})$ is an integer number we get:

$$\lceil t/d \rceil \leq \lceil numones(\{g(i) \ : \ i \in [t]\}) \rceil = numones(\{g(i) \ : \ i \in [t]\}).$$

□

3.2.4. Manipulating big sums in $\boldsymbol{VTC}^0$

We need to prove basic properties of summation (having a non-constant number of summands) like commutativity, associativity, distributivity, substitution in big sums, rearranging etc., in $\boldsymbol{VTC}^0$, to be able to carry out basic calculations in the theory. As a consequence of this section we will be able to freely derive inequalities and equalities between big summations (using rearranging, substitutions of equals, etc.) in $\boldsymbol{VTC}^0$.

Since big sums of rational numbers will be considered only when the rational numbered that are summed all have the same denominator (see Section 3.1.2), the proposition proved below is applicable also to big sums of rational numbers.

Proposition 3.22 (Basic properties of sums in $\boldsymbol{VTC}^0$)**.** *In what follows we consider the theory $\boldsymbol{VTC}^0$ over an extended language (including possibly new Σ_1^B-definable function symbols in $\boldsymbol{VTC}^0$ and their defining axioms). The function $f(i)$ is a number function symbol mapping to the rationals or naturals (possibly with additional undisplayed parameters). The theory $\boldsymbol{VTC}^0$ proves the following statements:*

Substitution: *Assume that $u(i), v(i)$ are two terms (possibly with additional undisplayed parameters), such that $u(i) = v(i)$ for all $i \leq n$, then*

$$\sum_{i=0}^{n} f(u(i)) = \sum_{i=0}^{n} f(v(i)).$$

Distributivity: *Assume that u is a term that does not contain the variable i, then*

$$u \cdot \sum_{i=0}^{n} f(i) = \sum_{i=0}^{n} u \cdot f(i).$$

Rearranging: *Assume that $I = \{0, \ldots, n\}$ and let $I_1, \ldots, I_k$ be a definable partition of I (specifically, the sets $I_1, \ldots, I_k$ are all Σ_0^B-definable in $\boldsymbol{VTC}^0$ and $\boldsymbol{VTC}^0$ proves that the I_j's form a partition of I). Then*

$$\sum_{i=0}^{n} f(i) = \sum_{j=1}^{k} \sum_{i \in I_j} f(i),$$

where $\sum_{i \in I_j} f(i)$ denotes the term $\sum_{i=0}^{|I_j|-1} f(\delta(i))$ where $\delta(i)$ is the function that enumerates (in ascending order) the elements in I_j.

Inequalities: *Let $g(i)$ be a number function mapping to the rationals or naturals (possibly with additional undisplayed parameters), such that $f(i) \leq g(i)$ for all $0 \leq i \leq n$, then*

$$\sum_{i=0}^{n} f(i) \leq \sum_{i=0}^{n} g(i).$$

Proof.
Substitution: When we work in the theory $\boldsymbol{VTC}^0$ we implicitly assume that we have equality axioms stating that if $t = t'$, for any two terms t, t', then $F(t) = F(t')$, for any function F (including functions F that are from

the extended language of $\boldsymbol{VTC}^0$). Since we assume that $f(i)$ is a Σ_1^B-definable number function in $\boldsymbol{VTC}^0$, the function $g(n) := \sum_{i=0}^{n} f(i)$ is also Σ_1^B-definable in $\boldsymbol{VTC}^0$, and so we also have the equality axiom for $g(n)$. Thus, if $u(i) = v(i)$, for any $i \le n$, then we can prove also $g(u(n)) = g(v(n))$.

Distributivity: This is proved simply by induction on n. We omit the details.

Rearranging: Because $I_1, \dots, I_k$ are Σ_0^B-definable sets in $\boldsymbol{VTC}^0$ we can define the family of sequences $S_1, \dots, S_k$, each of length $n+1$, such that

$$S_j[i] := \begin{cases} f(i), & i \in I_j \ ; \\ 0, & \text{otherwise.} \end{cases}$$

The theory $\boldsymbol{VTC}^0$ proves, by induction on n, that

$$\sum_{j=1}^{k} \sum_{i=0}^{n} S_j[i] = \sum_{i=0}^{n} f(i).$$

For any $j = 1, \dots, k$, we can Σ_1^B-define in $\boldsymbol{VTC}^0$ the function $\delta_j : \{0, \dots, |I_j| - 1\} \to \{0, \dots, n\}$ such that $\delta_j(\ell) = i$ iff i is the $(\ell+1)$th element in I_j (when the elements in I_j are ordered in ascending order). In other words, the δ_j's functions enumerate the elements in I_j.

We can now prove in $\boldsymbol{VTC}^0$ that

$$\sum_{i=0}^{n} S_j[i] = \sum_{i=0}^{|I_j|-1} f(\delta_j(i)),$$

from which, by Substitution (proved above), we can prove:

$$\sum_{i=1}^{k} \sum_{i=0}^{n} S_j[i] = \sum_{i=1}^{k} \sum_{i=0}^{|I_j|-1} f(\delta_j(i)).$$

Inequalities: This can be proved in $\boldsymbol{VTC}^0$ simply by induction on n. We omit the details. □

All the equalities and inequalities which contain big summations that we will derive in the theory, can be proved using Proposition 3.22. We shall not state this explicitly in the text, but continue to derive such equalities and inequalities freely.

3.2.5. The relation between $\boldsymbol{VTC}^0$ and $\boldsymbol{TC}^0$-Frege

In this section we show how one can translate a Σ_0^B formula φ into a family of propositional formulas $[\![\varphi]\!]$. We then state the theorem showing that if the universal closure of a Σ_0^B formula φ is provable in $\boldsymbol{VTC}^0$ then the propositional translation $[\![\varphi]\!]$ has a polynomial-size proof in $\boldsymbol{TC}^0$-Frege.

We will, however, only explain the translation for $\mathbf{V}^0$-formulas, as this is all we need to do explicitly to obtain our result. For the complete translation, we refer the reader to [17] Chapter X.4.

Definition 3.23 (Propositional translation $[\![\cdot]\!]$ of Σ_0^B formulas)**.** *Let $\varphi(\vec{x}, \vec{X})$ be a Σ_0^B formula. The* propositional translation *of φ is a family*

$$[\![\varphi]\!] = \{[\![\varphi]\!]_{\vec{m};\vec{n}} \mid m_i, n_i \in \mathbb{N}\}$$

of propositional formulas in variables $p_j^{X_i}$ for every $X_i \in \vec{X}$. The intended meaning is that $[\![\varphi]\!]$ is a valid family of formulas if and only if the formula

$$\forall\vec{x}\forall\vec{X}\left((\bigwedge |X_i| = \underline{n_i}) \to \varphi(\underline{\vec{m}}, \vec{X})\right)$$

is true in the standard model $\mathbb{N}_2$ of two-sorted arithmetic, where $\underline{n}$ denotes the nth numeral, for any $n \in \mathbb{N}$.

For given $\vec{m}, \vec{n} \in \mathbb{N}$ we define $[\![\varphi]\!]$ by induction on the size of the formula $[\![\varphi]\!]_{\vec{m};\vec{n}}$. We denote the value of a term t by $\mathsf{val}(t)$.

***Case 1**: Let $\varphi(\vec{x}, \vec{X})$ be an atomic formula.*

- *If $\varphi(\vec{x}, \vec{X})$ is $\top$ (or $\bot$), then $[\![\varphi]\!]_{\vec{m},\vec{n}} := \top$ (or $\bot$).*
- *If $\varphi(\vec{x}, \vec{X})$ is $X_i = X_i$, then $[\![\varphi]\!]_{\vec{m},\vec{n}} := \top$.*
- *If $\varphi(\vec{x}, \vec{X})$ is $X_i = X_j$ for $i \neq j$, then (using the fact that $\mathbf{V}^0$contains the extensionality axiom* **SE***) instead of translating φ, we translate the $\mathbf{V}^0$-equivalent formula*

$$|X_i| = |X_j| \wedge \forall k \leq |X| \, (X_i(k) \leftrightarrow X_j(k))).$$

- *If $\varphi(\vec{x}, \vec{X})$ is $t_1(\vec{y}, |\vec{Y}|) = t_2(\vec{z}, |\vec{Z}|)$ for terms t_1, t_2, number variables $\vec{y}, \vec{z}$ and string variables $\vec{Y}, \vec{Z}$, where $\vec{y} \cup \vec{z} = \vec{x}$ and $\vec{Y} \cup \vec{Z} = \vec{X}$, and $\underline{\vec{m}^y}, \underline{\vec{m}^z}$ and $\underline{\vec{n}^Y}, \underline{\vec{n}^Z}$ denote the corresponding assignments of numerals $\underline{\vec{m}}, \underline{\vec{n}}$ to the $\vec{y}, \vec{z}$ and $\vec{Y}, \vec{Z}$ variables, respectively. Then*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := \begin{cases} \top & \text{if } \mathsf{val}(t_1(\underline{\vec{m}^Y}, \underline{\vec{n}^Y})) = \mathsf{val}(t_2(\underline{\vec{m}^Z}, \underline{\vec{n}^Z})) \text{ and} \\ \bot & \text{otherwise.} \end{cases}$$

- *If* $\varphi(\vec{x}, \vec{X})$ *is* $t_1(\vec{y}, |\vec{Y}|) \leq t_2(\vec{z}, |\vec{Z}|)$ *for terms* t_1, t_2*, number variables* $\vec{y}, \vec{z}$ *and string variables* $\vec{Y}, \vec{Z}$*, then*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := \begin{cases} \top & \textit{if } \mathsf{val}(t_1(\underline{\vec{m}^Y}, \underline{\vec{n}^Y})) \leq \mathsf{val}(t_2(\underline{\vec{m}^Z}, \underline{\vec{n}^Z})) \textit{ and} \\ \bot & \textit{otherwise.} \end{cases}$$

- *If* $\varphi(\vec{x}, \vec{X})$ *is* $X_i(t(\vec{x}, |\vec{X}|))$*, then*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := \bot \quad \textit{if } n_i = 0$$

and otherwise

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := \begin{cases} p^{X_i}_{\mathsf{val}(t(\underline{\vec{m}}, \underline{\vec{n}}))} & \textit{if } \mathsf{val}(t(\underline{\vec{m}}, \underline{\vec{n}})) < \underline{n_i - 1}, \\ \top & \textit{if } \mathsf{val}(t(\underline{\vec{m}}, \underline{\vec{n}})) = \underline{n_i - 1}, \\ \bot & \textit{if } \mathsf{val}(t(\underline{\vec{m}}, \underline{\vec{n}})) > \underline{n_i - 1}. \end{cases}$$

Case 2: *The formula* φ *is not atomic.*

- *If* $\varphi \equiv \psi_1 \wedge \psi_2$ *we let*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := [\![\psi_1]\!]_{\vec{m},\vec{n}} \wedge [\![\psi_2]\!]_{\vec{m},\vec{n}}.$$

- *If* $\varphi \equiv \psi_1 \vee \psi_2$ *we let*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := [\![\psi_1]\!]_{\vec{m},\vec{n}} \vee [\![\psi_2]\!]_{\vec{m},\vec{n}}.$$

- *If* $\varphi \equiv \neg\psi$ *we let*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := \neg[\![\psi]\!]_{\vec{m},\vec{n}}.$$

- *If* $\varphi \equiv \exists y \leq t(\vec{x}, |\vec{X}|)\psi(y, \vec{x}, \vec{X})$ *then*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := \bigvee_{i=0}^{\mathsf{val}(t(\underline{\vec{m}}, \underline{\vec{n}}))} [\![\psi(\underline{i}, \vec{x}, \vec{X})]\!]_{\vec{m},\vec{n}}.$$

- *If* $\varphi \equiv \forall y \leq t(\vec{x}, |\vec{X}|)\psi(y, \vec{x}, \vec{X})$ *then*

$$[\![\varphi]\!]_{\vec{m},\vec{n}} := \bigwedge_{i=0}^{\mathsf{val}(t(\underline{\vec{m}}, \underline{\vec{n}}))} [\![\psi(\underline{i}, \vec{x}, \vec{X})]\!]_{\vec{m},\vec{n}}.$$

This concludes the translation for Σ^B_0 formulas.

Proposition 3.24 (Lemma VII.2.2. [17]). *For every Σ_0^B formula $\varphi(\vec{x}, \vec{X})$ there exists a constant $d \in \mathbb{N}$ and a polynomial $p(\vec{m}, \vec{n})$ such that for all $\vec{m}, \vec{n} \in \mathbb{N}$, the propositional translation $[\![\varphi(\vec{x}, \vec{X})]\!]_{\vec{m},\vec{n}}$ has depth at most d and size at most $p(\vec{m}, \vec{n})$.*

We can now state the relation between provability of an arithmetical statement φ in $\boldsymbol{VTC}^0$ to the provability of the family $[\![\varphi]\!]$ in $\boldsymbol{TC}^0$-Frege as follows.

Theorem 3.25 (Section X.4.3. [17]). *Let $\varphi(\vec{x}, \vec{X})$ be a Σ_0^B formula. Then, if $\boldsymbol{VTC}^0$ proves $\varphi(\vec{x}, \vec{X})$ then there is a polynomial-size family of $\boldsymbol{TC}^0$-Frege proofs of $[\![\varphi]\!]$.*

4. Feige-Kim-Ofek witnesses and the main formula

In this section we define the main formula we are going to prove in the theory. We are concerned with proofs of 3CNF formulas. Let us fix the following notation. By n we will denote the number of propositional variables $x_1, \dots, x_n$ and by m we will denote the number of clauses appearing in the 3CNF denoted $\mathbf{C} = \bigwedge_{\alpha=0}^{m-1} C_\alpha$. Each clause C_α is of the form $x_i^{\ell_1} \vee x_j^{\ell_2} \vee x_k^{\ell_3}$, for $\ell_1, \ell_2, \ell_3 \in \{0, 1\}$, where x_i^1 abbreviates x_i and x_i^0 abbreviates $\neg x_i$. A clause C_α is represented by the sequence $\big\langle i, j, k, \langle \ell_1, \ell_2, \ell_3 \rangle, \alpha \big\rangle$.

Formally, the defining Σ_0^B formula of the relation is:

$$\begin{aligned}&\textsc{Clause}(x, n, m) \leftrightarrow \exists i, j, k \leq n \exists \alpha < m \exists \ell_1, \ell_2, \ell_3 < 2 \\ &\quad (i > 0 \wedge j > 0 \wedge k > 0 \wedge \langle x \rangle_1^5 = i \wedge \langle x \rangle_2^5 = j \wedge \langle x \rangle_3^5 = k \wedge \langle x \rangle_4^5 = \langle \ell_1, \ell_2, \ell_3 \rangle \wedge \langle x \rangle_5^5 = \alpha).\end{aligned}$$

A 3CNF $\mathbf{C} \equiv \bigwedge_{\alpha=0}^{m-1} C_\alpha$ is represented by the sequence $(C_0, \dots, C_{m-1})$. Since m is not a constant, we use a string variable to code $\mathbf{C}$. The defining Σ_0^B formula of this relation is:

$$\mathrm{3CNF}(\mathbf{C}, n, m) \leftrightarrow \forall \alpha < m \left(\textsc{Clause}(\mathbf{C}[\alpha], n, m) \wedge \langle \mathbf{C}[\alpha] \rangle_5^5 = \alpha \right).$$

For a number variable x, we Σ_0^B-define $\textsc{Even}(x)$ by the formula $\exists y \leq x (2 \cdot y = x)$ (meaning that x is an even number). Accordingly, we define $\textsc{Odd}(x)$ by $\neg\textsc{even}(x)$.

For some clause C and a string variable A (interpreted as a Boolean assignment), we Σ_0^B-define the following predicate, stating that C is not satisfied under the assignment A:

$$\begin{aligned}\textsc{NotSAT}(C, A) \equiv & \exists i, j, k \leq n \\ & \qquad \big(\langle C\rangle^5_1 = i \wedge (A(i) \leftrightarrow \langle\langle C\rangle^5_4\rangle^3_1 = 0)\big) \\ & \qquad \wedge \big(\langle C\rangle^5_2 = j \wedge (A(j) \leftrightarrow \langle\langle C\rangle^5_4\rangle^3_2 = 0)\big) \\ & \qquad \wedge \big(\langle C\rangle^5_3 = k \wedge (A(k) \leftrightarrow \langle\langle C\rangle^5_4\rangle^3_3 = 0)\big)\,.\end{aligned}$$

We need the following notations and definitions to facilitate the formalization of certain sets and objects:

Notation:

1. When considering a *set* of clauses, then a clause in $\mathbf{C}$ will be referred to only by its index $0 \leq \alpha < m$. Thus, a set of clauses from $\mathbf{C}$ is a set of natural numbers less than m.
2. A set of literal positions from $\mathbf{C}$ will be coded as a set of numbers $\langle \alpha, b\rangle$, where $0 \leq \alpha < m$ is the index of a clause in $\mathbf{C}$ and $b = 1, 2, 3$ is the index of a literal in the clause.
3. For $0 \leq \alpha < m$ and $\varepsilon = 0, 1$ and a sequence S of 3-clauses we define $\mathsf{LitPos}(S, \alpha, \varepsilon)$ to be the string function that outputs the set of (positions of) literals x^ε_α in S. In other words, we have:

   $$\mathsf{LitPos}(S, \alpha, \varepsilon) := \big\{\langle j, \ell\rangle \;:\; j < \mathit{length}(S) \wedge \ell \leq 3 \wedge \langle S[j]\rangle^5_\ell = \alpha \;\wedge\; \langle\langle S[j]\rangle^5_4\rangle^3_\ell = \varepsilon\big\}\,.$$

4. Let $\mathsf{satLit}(A, \mathbf{C})$ be the string function that outputs the set of all literal positions in $\mathbf{C}$ that are satisfied by A.
5. The function $\mathsf{Lit}(C, i)$ returns the ith literal x^ε_j of the clause C, for $i = 1, 2, 3$, in the form of a pair $\langle j, \varepsilon\rangle$.
6. If the literals of a clause are not all true or not all false under A, then we say that the clause is satisfied as NAE (standing for "not all equal") by A. We can easily Σ^B_0-define the predicate $\textsc{SatL}(z, A)$, stating that the literal z is satisfied by the assignment A in $\boldsymbol{VTC}^0$. Let:

   $$\mathrm{NAE}(C, A) \;\leftrightarrow\; \textsc{Clause}(C) \wedge \bigvee_{i=1,2,3} \textsc{SatL}\left(\mathsf{Lit}(C, i), A\right) \wedge \bigvee_{i=1,2,3} \neg\textsc{SatL}\left(\mathsf{Lit}(C, i), A\right)$$

   be the Σ^B_0 relation that states that the assignment A satisfies the 3-clause C as NAE. Let $\mathsf{satNAE}(A, \mathbf{C})$ be the string function that outputs the set of clauses in $\mathbf{C}$ that are satisfied as NAE by A.

The functions $\mathsf{LitPos}(S, i, \varepsilon)$, $\mathsf{satLit}(A, \mathbf{C})$ and $\mathsf{satNAE}(A, \mathbf{C})$ above are all $\mathbf{AC}^0$-reducible to the language $\mathcal{L}^2_A$ and so we can assume that we have these functions (along with their defining axioms) in $\boldsymbol{VTC}^0$ (see Section 3.1.2). All the functions in this section will be $\mathbf{AC}^0$-reducible to $\mathcal{L}_A^2 \cup \{numones\}$, and all the relations in this section will have Σ_0^B definitions in the language $\mathcal{L}_A^2$ extended to include both our new function symbols and *numones*.

Definition 4.1 (Even k-tuple)**.** *For any given k, a sequence S of k clauses is an* even k-tuple *iff every variable appears an even number of times in the sequence. Formally, the predicate is denoted* $\mathrm{TPL}(S,k)$*:*

$$\begin{aligned}\mathrm{TPL}(S,k) \ \leftrightarrow\ & \mathit{length}(S) = k \wedge \\ & \forall i \leq n,\ \textsc{Even}\left(\mathit{numones}(\mathsf{LitPos}(S,i,0)) + \mathit{numones}(\mathsf{LitPos}(S,i,1))\right). \end{aligned} \tag{20}$$

Observe that if S is an even k-tuple then k is even (since the total number of variable occurrences N is even, by assumption that each variable occurs an even number of times; and $k = N/3$, since each clause has three variables).

Definition 4.2 (Inconsistent k-tuple)**.** *An even k-tuple is said to be* inconsistent *if the total number of negations in its clauses is odd. Formally, the predicate is denoted by* $\mathrm{ITPL}(S,k)$*:*

$$\mathrm{ITPL}(S,k) \leftrightarrow \mathrm{TPL}(S,k) \wedge \textsc{Odd}\left(\sum_{i=1}^{n} \mathit{numones}(\mathsf{LitPos}(S,i,1))\right).$$

Definition 4.3 (The imbalance $\textsc{Imb}(S,y)$)**.** *For a 3CNF S we define the function* i-imbalance $\mathsf{iImb}(S,i)$ *to be the absolute value of the difference of negated occurrences of x_i and non-negated occurrences of x_i in the 3CNF S (where $x_1,\dots,x_n$ are considered to be all the variables in S). It is defined simply by the term:*

$$\mathsf{iImb}(S,i) := \mathit{abs}(\mathit{numones}(\mathsf{LitPos}(S,i,0)) - \mathit{numones}(\mathsf{LitPos}(S,i,1))).$$

For a 3CNF S, the predicate imbalance of S, *denoted* $\textsc{Imb}(S,y)$, *is true iff y equals the sum over the i-imbalances of all the variables, that is:*

$$\textsc{Imb}(S,y) \leftrightarrow y = \sum_{i=1}^{n} \mathsf{iImb}(S,i).$$

Definition 4.4 ((t,k,d)-collection)**.** *A* (t,k,d)-collection $\mathscr{D}$ *of a 3CNF* $\mathbf{C}$ *with m clauses is an array (coded as in Definition 3.10) of t many inconsistent k-tuples, which contain only clauses from* $\mathbf{C}$*, and each clause appears* *in at most d many such inconsistent k-tuples. The predicate is denoted* $\textsc{Coll}(t, k, d, \mathbf{C}, \mathscr{D})$ *and is defined by the following formula:*

$$\begin{aligned}&\mathit{length}(\mathscr{D}) = t \wedge\\ &\quad \forall i < t\, \mathrm{ITPL}(\mathscr{D}^{[i]}, k) \wedge\\ &\quad \forall i < t \forall \ell < k \exists j < |\mathbf{C}|\, (\mathscr{D}^{[i]}[\ell] = \mathbf{C}[j]) \wedge\\ &\quad \forall j < |\mathbf{C}| \sum_{i=0}^{t-1} \sum_{\ell=0}^{k-1} \chi_{=}(\langle \mathscr{D}^{[i]}[\ell] \rangle_5^5, j) \le d.\end{aligned}$$

Definition 4.5 ($\textsc{Mat}(M, \mathbf{C})$)**.** *We define the predicate* $\textsc{Mat}(M, \mathbf{C})$ *that holds iff M is an $n \times n$ rational matrix such that M_{ij} equals $\frac{1}{2}$ times the number of clauses in $\mathbf{C}$ where x_i and x_j appear with a different polarity minus $\frac{1}{2}$ times the number of clauses where they appear with the same polarity. More formally, we have*

$$M_{ij} := \sum_{k=0}^{m-1} E_{ij}^{(k)}, \qquad \textit{for any } i, j \in [n], \tag{21}$$

where $E_{ij}^{(k)}$ corresponds to the kth clause in $\mathbf{C}$ as follows:

$$E_{ij}^{(k)} := \begin{cases} \frac{1}{2}, & x_i^{\varepsilon_i}, x_j^{\varepsilon_j} \in \mathbf{C}[k] \textit{ and } \varepsilon_i \neq \varepsilon_j, \textit{ for some } \varepsilon_i, \varepsilon_j \in \{0,1\} \textit{ and } i \neq j;\\ -\frac{1}{2}, & x_i^{\varepsilon_i}, x_j^{\varepsilon_j} \in \mathbf{C}[k] \textit{ and } \varepsilon_i = \varepsilon_j, \textit{ for some } \varepsilon_i, \varepsilon_j \in \{0,1\} \textit{ and } i \neq j;\\ 0, & \textit{otherwise.}\end{cases} \tag{22}$$

Note that $E_{ij}^{(k)}$ is definable by a Σ_0^B formula (in $\mathcal{L}_A^2$), and so $\textsc{Mat}(M, \mathbf{C})$ is a Σ_0^B-definable relation in $\boldsymbol{VTC}^0$.

Finally, we need a predicate $\textsc{EigValBound}(M, \vec{\lambda}, V)$ that ensures that $\vec{\lambda}$ is a collection of n rational approximations of the eigenvalues of the matrix M and that V is a rational matrix whose rows are rational approximations of the eigenvectors of M (where the ith row in V is the approximation of the approximate eigenvector λ_i). For the sake of readability we defer the formal definition of the predicate $\textsc{EigValBound}(M, \vec{\lambda}, V)$ and all the lemmas that relate to it, including the proofs in the theory making use of this predicate, to Section 6.

We can now state the main formula that we are going to prove in $\boldsymbol{VTC}^0$. It says that if the Feige-Kim-Ofek witness fulfills the inequality $t > \frac{d \cdot (I + \lambda n)}{2} + c' d/n$, where c' is a sufficiently large constant number term, then there exists a clause in $\mathbf{C}$ that is not satisfied by any assignment A (one can think of all the free variables in the formula as universally quantified):

Definition 4.6 (The main formula)**.** *The* main formula *is the following formula ($\vec{\lambda}$ denotes n distinct number parameters $\lambda_1, \ldots, \lambda_n$):*

$$\Big(3\mathrm{CNF}(\mathbf{C}, n, m) \wedge \textsc{Coll}(t, k, d, \mathbf{C}, \mathscr{D}) \wedge \textsc{Imb}(\mathbf{C}, I) \wedge \textsc{Mat}(M, \mathbf{C}) \wedge$$

$$\textsc{EigValBound}(M, \vec{\lambda}, V) \wedge \lambda = \max\{\lambda_1, \ldots, \lambda_n\} \wedge t > \frac{d \cdot (I + \lambda n)}{2} + \frac{c'd}{n}\Big)$$

$$\longrightarrow \exists i < m\, \textsc{NotSAT}(\mathbf{C}[i], A).$$

5. Proof of the main formula

In this section we prove the key theorem, stating that the theory $\boldsymbol{VTC}^0$ proves the main formula (Definition 4.6). This means that the theory proves the *correctness* of the FKO witness (namely, that its existence implies unsatisfiability). The formal proof in the theory is quite long and detailed. To facilitate the understanding of this formal proof (given in Section 5.2), we present a slightly less formal proof of the correctness of the FKO witnesses in what follows.

5.1. Informal proof of the main formula

Assume by way of contradiction that the premise of the implication in the Main Formula holds and that there is an assignment $A \in \{0,1\}^n$ (construed as a string variable of length n) that satisfies every clause in $\mathbf{C}$. Recall that $\mathsf{satLit}(A, \mathbf{C})$ is the string function that outputs the set of all positions of literals in $\mathbf{C}$ that are satisfied by A and that $\mathsf{satNAE}(A, \mathbf{C})$ is the string function that returns the set of all clauses in $\mathbf{C}$ that are satisfied as NAE by A.

Lemma 5.1. *(In $\boldsymbol{VTC}^0$) numones*$(\mathsf{satLit}(A, \mathbf{C})) \leq \frac{3m+I}{2}$.

This lemma is proved by basic counting in $\boldsymbol{VTC}^0$.

We now bound the number of clauses in $\mathbf{C}$ that contain exactly two literals satisfied by A.

Lemma 5.2. *(In $\boldsymbol{VTC}^0$) Assume the premise of the Main Formula and let h be the number of clauses in $\mathbf{C}$ that contain exactly two literals satisfied by A. Then*

$$h \le \frac{3m+I}{2} - 3m + 2 \cdot \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C}))\,.$$

Similarly to Lemma 5.1, Lemma 5.2 is proved by basic counting in $\boldsymbol{VTC}^0$ as follows. Let ℓ, h and g be the number of clauses in $\mathbf{C}$ that have precisely one, two and three literals satisfied under A, respectively. By Lemma 5.1 we have $(3m+I)/2 \ge 1 \cdot \ell + 2 \cdot h + 3 \cdot g$. Let $f = \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C}))$. Then by definition, $\ell + 2h = 2f - (f-h) = f + h$. Also, $g = m - f$ (since by assumption every clause has at least one literal set to true under A). We therefore get

$$\begin{aligned}(3m+I)/2 &\ge \ell + 2h + 3g \\ &= f + h + 3m - 3f \\ &= h + 3m - 2f,\end{aligned}$$

which implies that $h \le (3m+I)/2 - 3m + 2f$, concluding the lemma.

We now wish to provide an upper bound on the number of clauses in $\mathbf{C}$ that can be satisfied as NAE by the assignment A, that is, $\mathit{numones}(\mathsf{satNAE}(A, \mathbf{C}))$. First we need the technical claim below.
Notation: For an assignment A we define its associated vector $\mathbf{a} \in \{-1,1\}^n$ such that $\mathbf{a}(i) = 1$ if $A(i) = 1$ and $\mathbf{a}(i) = -1$ if $A(i) = 0$.

Claim 5.3. *(In $\boldsymbol{VTC}^0$) Assume the premise of the Main Formula holds and let $f = \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C}))$. Then, $\mathbf{a}^t M \mathbf{a} = f - 3(m-f) = 4f - 3m$.*

Proof of claim: First note that by symmetry of M we have $\mathbf{a}^t M \mathbf{a} = \sum_{i,j\in[n]} \mathbf{a}(i)\mathbf{a}(j)M_{ij} = \sum_{i<j\in[n]} 2\mathbf{a}(i)\mathbf{a}(j)M_{ij}$.

The proof follows by the definition of the matrix M. Recall that $M_{ij} := \sum_{k<m} E_{ij}^{(k)}$, where $E_{ij}^{(k)}$ is the contribution of clause C_k to M_{ij} (Definition 4.5). Then, $\sum_{i<j\in[n]} 2\mathbf{a}(i)\mathbf{a}(j)M_{ij} = \sum_{i<j\in[n]} 2\mathbf{a}(i)\mathbf{a}(j)\sum_{k<m} E_{ij}^{(k)} = \sum_{k<m}\sum_{i<j\in[n]} 2\mathbf{a}(i)\mathbf{a}(j)E_{ij}^{(k)}$.

Fix some $0 \le k < m$, and consider $\sum_{i<j\in[n]} 2\mathbf{a}(i)\mathbf{a}(j)E_{ij}^{(k)}$. If the variable x_i does not occur in clause C_k, then by definition, $E_{ij}^{(k)} = 0$, for all j. Thus, considering all six possible cases where $2\mathbf{a}(i)\mathbf{a}(j)E_{ij}^{(k)}$ is nonzero, it is not hard to prove that $2\mathbf{a}(i)\mathbf{a}(j)E_{ij}^{(k)}$ is 1 if C_k is satisfied as NAE by A, and

otherwise it is -3. Thus, we can prove that $\mathbf{a}^t M\mathbf{a} = 1 \cdot f - 3(m-f) = 4f - 3m$. $\blacksquare_{\text{Claim}}$

Lemma 5.4 (In $\boldsymbol{VTC}^0$)**.** *If the premise of the Main Formula holds then*

$$\mathit{numones}(\mathsf{satNAE}(A, \mathbf{C})) \leq (\lambda n + 3m)/4 + \Theta(1/n)\,.$$

To prove this lemma we reason as follows: let $f = \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C}))$. Then, $4f - 3m = \mathbf{a}^t M\mathbf{a}$ by the previous claim. Thus, $f = (\mathbf{a}^t M\mathbf{a} + 3m)/4$. Hence, it suffices to prove the following basic spectral inequality:

Lemma 5.5 (Main spectral bound)**.** *(In $\boldsymbol{VTC}^0$) If* $\textsc{EigValBound}(M, \vec{\lambda}, V)$ *holds, then for any assignment A to n variables:*

$$\mathbf{a}^t M\mathbf{a} \leq \lambda n + \Theta(1/n)\,.$$

We can now finish the proof of the key theorem. In $\boldsymbol{VTC}^0$ (and assuming the premise of the Main Formula), let h be the number of clauses in $\mathbf{C}$ that contain exactly two literals satisfied by A. By Lemmata 5.2 and 5.4, we have:

$$\begin{aligned} h &\leq \frac{3m+I}{2} - 3m + 2 \cdot \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C})) \\ &\leq \frac{3m+I}{2} - 3m + \frac{3m+\lambda n}{2} + \Theta(1/n) = \frac{I+\lambda n}{2} + \Theta(1/n)\,. \end{aligned} \tag{23}$$

Since we assumed that A satisfies $\mathbf{C}$, then every clause in $\mathbf{C}$ has at least one literal satisfied by A. Thus, the clauses in $\mathbf{C}$ that are not satisfied as 3XOR by A are precisely the clauses that have exactly two literals satisfied by A. By (23), the number of clauses that have exactly two literals satisfied by A is at most $\frac{I+\lambda n}{2} + \Theta(1/n)$. We now use our witness, assumed to exist in the premise of the Main Formula, to show that:

Lemma 5.6 (In $\boldsymbol{VTC}^0$)**.** *(Assuming the premise of the Main Formula) the number of clauses in $\mathbf{C}$ that are not satisfied as 3XOR by A is at least $\lceil t/d \rceil$.*

This concludes the key theorem, since the number of clauses in $\mathbf{C}$ not satisfied as 3XOR by A is at most $\frac{I+\lambda n}{2} + \Theta(1/n)$, and so by Lemma 5.6 we get that $t \leq d \cdot \left\lceil \frac{t}{d} \right\rceil \leq d \cdot \frac{I+\lambda n}{2} + \Theta(d/n)$, which contradicts our assumption (in the Main Formula) that $t > \frac{d(I+\lambda n)}{2} + \frac{c'd}{n}$, for a sufficiently large constant term c'.

To prove Lemma 5.6 we need the following:

Lemma 5.7. *(In $\boldsymbol{VTC}^0$) If S is an inconsistent (even) k-tuple, then for every assignment A to its variables there exists a clause in S that is* not *satisfied as 3XOR.*

The proof of this lemma is again by a basic counting argument, as follows: assume by way of contradiction that all the clauses in the k-tuple are satisfied as 3XOR under A. We consider the sum modulo 2 of all literals in the k-tuple under A. First we consider this sum by summing over clauses; since k is even, and by assumption all clauses are satisfied as 3XOR under A, summing over clauses evaluates to 0.

Now we consider summing the literals under A over variables. It is not hard to show that since every number occurs even number of times in the k-tuple and since the number of negative literals (and hence also the number of positive literals) is *odd*, summing the literals over variables evaluates to 1, modulo 2, and we obtain a contradiction.

It is important to note that the reason why we can carry out the above argument in $\boldsymbol{VTC}^0$ is that we can define (and reason about) *the parity of big sums of numbers.*

5.2. The formal proof in the theory

We now prove the key theorem:

Theorem 5.8 (Key)**.** *The theory $\boldsymbol{VTC}^0$ proves the main formula (Definition 4.6).*

The proof of this theorem follows the (informal) sketch given in Section 5.1 above.

Proof. We reason inside $\boldsymbol{VTC}^0$. Assume by way of contradiction that the premise of the implication in the main formula holds and that there is an assignment $A \in \{0,1\}^n$ (construed as a string variable of length n) that satisfies every clause in $\mathbf{C}$. Recall that $\mathsf{satLit}(A, \mathbf{C})$ is the set of all literal positions that are satisfied by A.

The following lemma is a reiteration of Lemma 5.1 in Section 5.1:

Lemma 5.9. *(Assuming the premise of the main formula) the theory $\boldsymbol{VTC}^0$ proves:*

$$numones(\mathsf{satLit}(A, \mathbf{C})) \leq \frac{3m + I}{2}.$$

Proof. First observe that for any assignment A and any $1 \leq i \leq n$ the set of satisfied literals of x_i is defined by $\mathsf{LitPos}(\mathbf{C}, i, A(i))$. Therefore, the sets

$\mathsf{LitPos}(\mathbf{C},1,A(1)),\ldots,\mathsf{LitPos}(\mathbf{C},n,A(n))$ form a partition of $\mathsf{satLit}(A,\mathbf{C})$ (provably in $\boldsymbol{VTC}^0$), and thus by Proposition 3.17, $\boldsymbol{VTC}^0$ proves that

$$numones(\mathsf{satLit}(A,\mathbf{C})) = \sum_{i=1}^{n} numones(\mathsf{LitPos}(\mathbf{C},i,A(i))). \tag{24}$$

By (24) we get

$$numones(\mathsf{satLit}(A,\mathbf{C})) \le \sum_{i=1}^{n} \max\{numones(\mathsf{LitPos}(\mathbf{C},i,0)), numones(\mathsf{LitPos}(\mathbf{C},i,1))\}. \tag{25}$$

For any $1 \le i \le n$, define the term

$$\mathsf{LitPos}(\mathbf{C},i) := \mathsf{LitPos}(\mathbf{C},i,0) \cup \mathsf{LitPos}(\mathbf{C},i,1).$$

Then by

$$\frac{\mathsf{iImb}(\mathbf{C},i) + numones(\mathsf{LitPos}(\mathbf{C},i))}{2} = \frac{\mathsf{iImb}(\mathbf{C},i) + numones(\mathsf{LitPos}(\mathbf{C},i,0)) + numones(\mathsf{LitPos}(\mathbf{C},i,1))}{2},$$

and since, by Definition 4.3, $\mathsf{iImb}(\mathbf{C},i) = abs\,(numones(\mathsf{LitPos}(\mathbf{C},i,0)) - numones(\mathsf{LitPos}(\mathbf{C},i,1)))$, the theory $\boldsymbol{VTC}^0$ proves that for any $1 \le i \le n$:

$$\max\{numones(\mathsf{LitPos}(\mathbf{C},i,0)), numones(\mathsf{LitPos})(\mathbf{C},i,1)\} = \frac{\mathsf{iImb}(\mathbf{C},i) + numones(\mathsf{LitPos}(\mathbf{C},i))}{2}. \tag{26}$$

Claim 5.10. *(Assuming the premise of the main formula) the theory* $\boldsymbol{VTC}^0$ *proves:*

$$\sum_{i=1}^{n} \frac{\mathsf{iImb}(\mathbf{C},i) + numones(\mathsf{LitPos}(\mathbf{C},i))}{2} = \frac{I+3m}{2}.$$

Proof of claim: First recall the definition of imbalance (Definition 4.3) $I = \sum_{i=1}^{n} \mathsf{iImb}(\mathbf{C},i)$. Thus it remains to prove that $\sum_{i=1}^{n} numones(\mathsf{LitPos}(\mathbf{C},i)) = 3m$. For this, note that $\mathsf{LitPos}(\mathbf{C},i)$, for $i = 1,\ldots,n$, partition the set of all literal positions in $\mathbf{C}$. In other words, we can prove that: (i) if H is the set of all literal positions in $\mathbf{C}$ (this set is clearly Σ_0^B-definable in $\boldsymbol{VTC}^0$) then $H = \cup_{i=1}^{n} \mathsf{LitPos}(\mathbf{C},i)$; and (ii)

$\mathsf{LitPos}(\mathbf{C}, i) \cap \mathsf{LitPos}(\mathbf{C}, j) = \emptyset$, for all $1 \le i \ne j \le n$. Therefore, by Proposition 3.17 we can prove that:

$$numones(H) = \sum_{i=1}^{n} numones(\mathsf{LitPos}(\mathbf{C}, i)). \tag{27}$$

Now, the set H of all literal position in $\mathbf{C}$ can be partitioned (provably in $\boldsymbol{VTC}^0$) by the sets $T_1, \ldots, T_m$, where each T_j, for $0 \le j < m$, is the set of the three literals in the jth clause in $\mathbf{C}$. Thus, again by Proposition 3.17, we can prove that $numones(H) = 3m$. By (27) we therefore have

$$\sum_{i=1}^{n} numones(\mathsf{LitPos}(\mathbf{C}, i)) = 3m.$$

■Claim

We conclude that:

$$\begin{aligned}
&numones(\mathsf{satLit}(A, \mathbf{C})) \\
&\quad \le \sum_{i=1}^{n} \max\{numones(\mathsf{LitPos}(\mathbf{C}, i, 0)), numones(\mathsf{LitPos}(\mathbf{C}, i, 1))\} && \text{(by (25))} \\
&\quad = \sum_{i=1}^{n} \frac{\mathsf{iImb}(\mathbf{C}, i) + \mathsf{LitPos}(\mathbf{C}, i)}{2} && \text{(by (26))} \\
&\quad = \frac{I + 3m}{2}. && \text{(by Claim 5.10).}
\end{aligned}$$

□

We now bound the number of clauses in $\mathbf{C}$ that contain exactly two literals satisfied by A. We say that a 3-clause is satisfied by a given assignment as NAE (which stands for *not all equal*) if the literals in the clause do not all have the same truth values. That is, if either exactly one or exactly two literals in the clause are satisfied by the assignment.

Recall that $\mathsf{satNAE}(A, \mathbf{C})$ is the function that returns the set of all clauses (formally, indices $< m$) that are satisfied as NAE by A.

The following lemma is a reiteration of Lemma 5.2 in Section 5.1:

Lemma 5.11. *(Assuming the premise of the main formula) the theory $\boldsymbol{VTC}^0$ proves: let h be the number of clauses in $\mathbf{C}$ that contain exactly two literals satisfied by A. Then*

$$h \le \frac{3m + I}{2} - 3m + 2 \cdot numones(\mathsf{satNAE}(A, \mathbf{C}))\,.$$

Proof. For $i = 0, 1, 2, 3$, let B_i be the set of clauses in $\mathbf{C}$ that contain exactly i literals satisfied by A. For $i = 0, 1, 2, 3$, let F_i be the string function that maps a clause (index) C to the set of literal positions that are satisfied by A in case there are exactly i such literals and to the empty set otherwise:

$$F_i(j) = \begin{cases} \{l_1, \dots, l_i\}, & \text{if } j \in B_i \ ; \\ \emptyset, & \text{otherwise} \end{cases}$$

(where a literals l_k is coded, as before, by the pair $\langle a, b\rangle$ for a an index of a clause in $\mathbf{C}$ and b the position of the literal in the clause). Every such function F_i is Σ_0^B-defined in $\boldsymbol{VTC}^0$. We also Σ_0^B-define the image of F_i as follows:

$$\mathrm{Img}(F_i) := \{x : \ \exists y < m \, (F_i(y))(x)\}.$$

Claim 5.12. *(Assuming the premise of the main formula) If A is a satisfying assignment for $\mathbf{C}$, then the theory $\boldsymbol{VTC}^0$ proves:*

$$numones(\mathsf{satLit}(A, \mathbf{C})) = \sum_{i=1}^{3} numones(\mathrm{Img}(F_i)).$$

Proof of claim: In light of Proposition 3.17, it suffices to prove that $\mathsf{satLit}(A, \mathbf{C})$ is partitioned by $\mathrm{Img}(F_1), \mathrm{Img}(F_2), \mathrm{Img}(F_3)$ (note that $\mathrm{Img}(F_0) = \emptyset$ by definition), in the sense that:

(i) $\mathsf{satLit}(A, \mathbf{C}) = \mathrm{Img}(F_1) \cup \mathrm{Img}(F_2) \cup \mathrm{Img}(F_3)$, and
(ii) $\mathrm{Img}(F_i) \cap \mathrm{Img}(F_j) = \emptyset$, for all $1 \leq i \neq j \leq 3$.

We prove (i): consider a literal $x \in \mathsf{satLit}(A, \mathbf{C})$, and let $x = \langle a, b\rangle$. We know that the clause C_a contains the literal x. Now, either zero, or one, or two of the remaining literals in C_a are satisfied by A. So x must be in either $F_1(a)$ or in $F_2(a)$ or in $F_3(a)$, respectively. Item (ii) is easy to prove by the definition of the F_i's. We omit the details. $\blacksquare_{\text{Claim}}$

Claim 5.13. *For any $i = 1, 2, 3$, $numones(\mathrm{Img}(F_i)) = i \cdot numones(B_i)$.*

Proof of claim: Fix some $i = 1, 2, 3$. We prove the claim by induction on the number of clauses $j < m$ (we can consider the sets B_i and the functions F_i having an additional parameter that determines until which clause to build the sets. That is, $B_i(z)$ is the set of clauses from 0 to z that have i literals satisfied by A; and similarly we add a parameter for the F_i's). In the base case $j = 0$ there is only one clause C_0. Depending on A we know how many literals in C_0 are satisfied by A. So $0 \in B_i$ iff i literals are satisfied by A in C_0 iff $\mathit{numones}(F_i(0)) = i = i \cdot 1 = i \cdot \mathit{numones}(B_i)$. The induction step is similar and we omit the details. $\blacksquare_{\text{Claim}}$

By Claim 5.12 and Claim 5.13 we get:

$$\begin{aligned}\mathit{numones}(\mathsf{satLit}(A,\mathbf{C})) &= \sum_{i=1,2,3} \mathit{numones}(\mathrm{Img}(F_i)) \\ &= \sum_{i=1,2,3} i \cdot \mathit{numones}(B_i)\,. \qquad (28)\end{aligned}$$

It is easy to show (in a similar manner to Claim 5.12) that $B_1 \cup B_2 \cup B_3 = \{0,\dots,m-1\}$ and $B_i \cap B_j = \emptyset$, for any $1 \le i \ne j \le 3$. From this, using Proposition 3.17, we get that $m = \mathit{numones}(B_1) + \mathit{numones}(B_2) + \mathit{numones}(B_3)$, and so:

$$\mathit{numones}(B_1) = m - \mathit{numones}(B_2) - \mathit{numones}(B_3)\,. \qquad (29)$$

Thus, by (28):

$$\begin{aligned}\mathit{numones}(\mathsf{satLit}(A,\mathbf{C})) &= m - \mathit{numones}(B_2) - \mathit{numones}(B_3) + 2 \cdot \mathit{numones}(B_2) + 3 \cdot \mathit{numones}(B_3) \\ &= m + 2 \cdot \mathit{numones}(B_3) + \mathit{numones}(B_2)\,,\end{aligned}$$

and so

$$\mathit{numones}(B_2) = \mathit{numones}(\mathsf{satLit}(A,\mathbf{C})) - m - 2 \cdot \mathit{numones}(B_3)\,. \qquad (30)$$

The set of clauses in $\mathbf{C}$ that are NAE satisfied by A (i.e., $\mathsf{satNAE}(A,\mathbf{C})$) is equal to the set of clauses having either one or two literals satisfied by A; the latter two sets are just B_1 and B_2, and since they are (provably in $\boldsymbol{VTC}^0$) disjoint we have (using also (29)):

$$\mathit{numones}(B_3) = m - (\mathit{numones}(B_1) + \mathit{numones}(B_2)) = m - \mathit{numones}(\mathsf{satNAE}(A,\mathbf{C}))\,.$$

Plugging this into (30), and using Lemma 5.9, we get:

$$\begin{aligned}\mathit{numones}(B_2) &= \mathit{numones}(\mathsf{satLit}(A,\mathbf{C})) - 3m + 2 \cdot \mathit{numones}(\mathsf{satNAE}(A,\mathbf{C})) \\ &\le \frac{3m+I}{2} - 3m + 2 \cdot \mathit{numones}(\mathsf{satNAE}(A,\mathbf{C})).\end{aligned}$$

This concludes the proof of Lemma 5.11 □

Notation:

1. The notation $\Theta(1/n)$ appearing inside formulas in the proof within the theory, stands for a term of the form b/n, for b a sufficiently large number constant and n a number variable (and where a rational number is encoded in the way described in Section 3.1.2).
2. Given two terms t and $f(n)$ in the language $\mathcal{L}^2_A$, where n is a number variable, we say that $\boldsymbol{VTC}^0$ *proves* $t = O(f(n))$, to mean that there exists some constant b (independent of n; namely, b is a term without variables in the language $\mathcal{L}^2_A$), such that $\boldsymbol{VTC}^0$ proves $t \le b \cdot f(n)$.

The following lemma provides an upper bound on the number of clauses in $\mathbf{C}$ that can be satisfied as NAE by the assignment A (it is a reiteration of Lemma 5.4 in Section 5.1):

Lemma 5.14. *(Assuming the premise of the main formula) the theory* $\boldsymbol{VTC}^0$ *proves:*

$$\mathit{numones}(\mathsf{satNAE}(A, \mathbf{C})) \le (n\lambda + 3m)/4 + \Theta(1/n).$$

The proof of this lemma involves a spectral argument. Carrying out this argument in the theory is fairly difficult because one has to work with rational approximations (as the eigenvalues and eigenvectors might be irrationals, and so undefined in the theory) and further the proof must be sufficiently constructive, in the sense that it would fit in the theory $\boldsymbol{VTC}^0$. We thus defer to a separate section (Section 6) all treatment of the spectral argument. Given the desired spectral inequality, we can prove Lemma 5.14—this is done in Section 5.4.

We can now finish the proof of the key theorem:

Concluding the proof of the theorem (Theorem 5.8). In $\boldsymbol{VTC}^0$ (and assuming the premise of the main formula), let h be the number of clauses in $\mathbf{C}$ that contain exactly two literals satisfied by A. We have:

$$\begin{aligned} h &\le \frac{3m+I}{2} - 3m + 2 \cdot \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C})) && \text{(by Lemma 5.11)} \\ &\le \frac{3m+I}{2} - 3m + \frac{3m+\lambda n}{2} + \Theta(1/n) && \text{(by Lemma 5.14)} \\ &= \frac{I+\lambda n}{2} + \Theta(1/n)\,. && (31) \end{aligned}$$

Since we assumed that A satisfies $\mathbf{C}$, then every clause in $\mathbf{C}$ has at least one literal satisfied by A. Thus, the clauses in $\mathbf{C}$ that are not satisfied as 3XOR by A are precisely the clauses that have exactly two literals satisfied by A. By (31), the number of clauses that have exactly two literals satisfied by A is at most $\frac{I+\lambda n}{2} + \Theta(1/n)$.

We now use Lemma 5.16 (proved in the next subsection) to prove the following reiteration of Lemma 5.6 from Section 5.1:

Lemma 5.15. *(Assuming the premise of the main formula) the theory* $\boldsymbol{VTC}^0$ *proves that the number of clauses in* $\mathbf{C}$ *that are not satisfied as 3XOR by* A *is at least* $\lceil t/d \rceil$*.*

Proof. Consider the collection $\textsc{Coll}(t, k, d, \mathbf{C}, \mathscr{D})$ in the premise of the main formula. Then, $\mathscr{D}$ is a sequence of t inconsistent k-tuples from $\mathbf{C}$, and every k-tuple in $\mathscr{D}$ intersects[8] in the same clause from $\mathbf{C}$ with at most $d-1$ other k-tuples. By Lemma 5.16, each of the t inconsistent k-tuples contains a clause which is unsatisfied as 3XOR by A. Since each such clause may appear in at most d inconsistent k tuples, using Proposition 3.20 the theory $\boldsymbol{VTC}^0$ proves that the total number of distinct clauses not satisfied as 3XOR by A is at least $\lceil t/d \rceil$. □

Using this lemma, we can finish the proof of the key Theorem 5.8, as follows: by Lemma 5.15 and the fact that the number of clauses in $\mathbf{C}$ that are not satisfied as 3XOR by A is at most $\frac{I+\lambda n}{2} + \Theta(1/n)$, we get

$$t = d \cdot \frac{t}{d} \leq d \cdot \left\lceil \frac{t}{d} \right\rceil \leq d \cdot \frac{I + \lambda n}{2} + \Theta(d/n)\,, \tag{32}$$

which contradicts our assumption (in the main formula) that $t > \frac{d(I+\lambda n)}{2} + \frac{c'd}{n}$, for c' a sufficiently large constant term c'. □

5.3. Formulas satisfied as 3XOR

Here we prove the missing lemma that was used in the proof of Lemma 5.15.

Notation: For a sequence S of k many 3-clauses, and for $0 \leq \alpha < k$, we denote the three variables in the clause $S[\alpha]$ by $x_{i_\alpha}, x_{j_\alpha}, x_{h_\alpha}$, and abbreviate $\langle\langle S[\alpha]\rangle^5_4\rangle^3_t$, which is the polarity of the tth variable in $S[\alpha]$, by ℓ^α_t, for $t = 1, 2, 3$. Thus, $x_i^{\ell^\alpha_1}, x_j^{\ell^\alpha_2}, x_h^{\ell^\alpha_3}$, are the three literals in $S[\alpha]$ and the values of $\neg A(i) \oplus \ell^\alpha_1, \neg A(j) \oplus \ell^\alpha_2, \neg A(h) \oplus \ell^\alpha_3$ are the values that A assigns to $x_i^{\ell^\alpha_1}, x_j^{\ell^\alpha_2}, x_h^{\ell^\alpha_3}$, respectively, where $\oplus$ is the XOR operator. We also abuse

[8] Where a clause is identified with its index $0, \ldots, m-1$ in $\mathbf{C}$, so that two identical clauses with a different index are considered as two different clauses.

notation and write $\neg A(i)$ inside a term to mean the *characteristic function* of the predicate $\neg A(i)$, that is, the function that returns 1 if $\neg A(i)$ is true, and 0 otherwise.

For a clause C and an assignment A the predicate $3\mathrm{XOR}(C,A)$ says that A satisfies exactly one or three of the literals in C. If we denote by x_i, x_j, x_h the three variables in C and by ℓ_1, ℓ_2, ℓ_3 their respective polarities, we have:

$$3\mathrm{XOR}(C,A) \quad \text{iff} \quad \neg A(i) \oplus \ell_1 + \neg A(j) \oplus \ell_2 + \neg A(h) \oplus \ell_3 = 1 \mod 2\,,$$

and formally the predicate 3XOR is Σ_0^B-definable by the following formula:

$$3\mathrm{XOR}(C,A) := \textsc{Odd}(\neg A(i) + \ell_1 + \neg A(j) + \ell_2 + \neg A(h) + \ell_3)\,.$$

The following lemma is a reiteration of Lemma 5.7 in Section 5.1:

Lemma 5.16. *The theory $\boldsymbol{VTC}^0$ proves that if S is an inconsistent (even) k-tuple, then for every assignment A to its variables there exists $\alpha < k$ such that A satisfies exactly zero or exactly two literals in the clause $S[\alpha]$. More formally, $\boldsymbol{VTC}^0$ proves:*

$$\forall A \le n\, \forall k \le n \forall S \le p(n)\, \exists \alpha < k\, (|A| = n \wedge \mathrm{ITPL}(S,k) \rightarrow \neg 3\mathrm{XOR}\,(S[\alpha], A))\ ,$$

for some (polynomial) term $p(\cdot)$.

Proof. We need the following claim:

Claim 5.17. *Let $f(y)$ be a number function definable in $\boldsymbol{VTC}^0$. Then $\boldsymbol{VTC}^0$ proves the following statements:*

1. $(\forall \alpha < k, \textsc{Odd}(f(\alpha))) \wedge \textsc{Even}(k) \rightarrow \textsc{Even}\left(\sum_{\alpha=0}^{k-1} f(\alpha)\right)$*;*
2. $(\forall \alpha < k, \textsc{Even}(f(\alpha))) \rightarrow \textsc{Even}\left(\sum_{\alpha=0}^{k-1} f(\alpha)\right)$*;*
3. $(\forall \alpha < k, \textsc{Odd}(f(\alpha))) \wedge \textsc{Odd}(k) \rightarrow \textsc{Odd}\left(\sum_{\alpha=0}^{k-1} f(\alpha)\right)$*.*

Proof of claim: Consider Item 1 (the other items are similar). The proof is by induction on k, showing that

$$((\forall \alpha < k \exists y(2y + 1 = f(\alpha))) \wedge \exists y(2y = k)) \rightarrow \exists y \sum_{\alpha=0}^{k-1} f(\alpha) = 2y\,,$$

and using the fact that $\mathbf{V}^0$ proves that $\textsc{Odd}(x) \leftrightarrow \exists y \le x(2y+1 = x)$ (e.g., by induction on x). We omit the details. $\blacksquare_{\text{Claim}}$

Now, assume by way of contradiction that A satisfies all the clauses in S as 3XORs. Thus, for any $\alpha < k$, if we define $f(\alpha) := \neg A(i_\alpha) + \ell_1^\alpha + \neg A(j_\alpha) + \ell_2^\alpha + \neg A(h_\alpha) + \ell_3^\alpha$, then $\textsc{Odd}(f(\alpha))$. Hence, because $\textsc{Even}(k)$, by Claim 5.17 we can prove that:

$$\sum_{\alpha=0}^{k-1} (\neg A(i_\alpha) \oplus \ell_1^\alpha + \neg A(j_\alpha) \oplus \ell_2^\alpha + \neg A(h_\alpha) \oplus \ell_3^\alpha) = 0 \mod 2. \tag{33}$$

Recall that every variable appears an even number of times in S. Thus, a variable has an odd number of negative appearances if and only if it has has an odd number of positive appearances. Let $I_0 \in \{0, \dots, n-1\}$ be the indices of variables having an even number of positive (and thus negative) appearances in S and let $I_1 = \{0, \dots, n-1\} \setminus I_0$ be the indices of variables having an odd number of positive (and thus negative) appearances in S. Thus, the left hand side of (33) can be written as follows (for $\varepsilon = 0, 1$, we denote by $x_i^\varepsilon(A)$ the truth value of the literal x_i^ε under A):

$$\begin{aligned}\sum_{i \in I_0} \Bigg(\underbrace{x_i^1(A) + \ldots + x_i^1(A)}_{\text{even times}} + \underbrace{x_i^0(A) + \ldots + x_i^0(A)}_{\text{even times}} \Bigg) + \\ \sum_{i \in I_1} \Bigg(\underbrace{x_i^1(A) + \ldots + x_i^1(A)}_{\text{odd times}} + \underbrace{x_i^0(A) + \ldots + x_i^0(A)}_{\text{odd times}} \Bigg) .\end{aligned} \tag{34}$$

Claim 5.18. *For any $i \in I_0$ (and any string variable A of size n) the theory $\boldsymbol{VTC}^0$ proves that*

$$\underbrace{x_i^1(A) + \ldots + x_i^1(A)}_{\text{even times}} + \underbrace{x_i^0(A) + \ldots + x_i^0(A)}_{\text{even times}}$$

is an even number.

Proof of claim: Reason in $\boldsymbol{VTC}^0$ as follows: assume that $A(i) = 0$. Then $x_i^1(A) = 0$ and $x_i^0(A) = 1$ and so by Claim 5.17 the sum of evenly many $x_i^1(A)$'s is even and the sum of evenly many $x_i^0(A)$'s is also even. The sum of two even numbers is even, and so we are done. (The case where $A(i) = 1$ is similar.) $\blacksquare_{\text{Claim}}$

By Claims 5.17 and 5.18, the theory $\boldsymbol{VTC}^0$ proves

$$\textsc{Even} \left(\sum_{i \in I_0} \Bigg(\underbrace{x_i^1(A) + \ldots + x_i^1(A)}_{\text{even times}} + \underbrace{x_i^0(A) + \ldots + x_i^0(A)}_{\text{even times}} \Bigg) \right) . \tag{35}$$

Similarly to the above claims we have:

Claim 5.19. *For any $i \in I_1$ (and any string variable A of size n) the theory $\boldsymbol{VTC}^0$ proves that*

$$\underbrace{x_i^1(A) + \ldots + x_i^1(A)}_{\text{odd times}} + \underbrace{x_i^0(A) + \ldots + x_i^0(A)}_{\text{odd times}}$$

is an odd number.

Since by assumption S is an inconsistent k-tuple, the number of negative literals is odd (Definition 4.2), and so (provably in $\boldsymbol{VTC}^0$) the number of variables that have an odd number of negative appearances must be odd, in other words, $|I_1|$ is odd. Therefore, by Claims 5.19 and 5.17, $\boldsymbol{VTC}^0$ proves:

$$\textsc{Odd}\left(\sum_{i \in I_1}\left(\underbrace{x_i^1(A) + \ldots + x_i^1(A)}_{\text{odd times}} + \underbrace{x_i^0(A) + \ldots + x_i^0(A)}_{\text{odd times}}\right)\right). \tag{36}$$

Since $\boldsymbol{VTC}^0$ proves both (35) and (36), $\boldsymbol{VTC}^0$ proves that (34) is odd, which contradicts (33). This implies that not all the clauses in S are satisfied as 3XOR by the assignment A. □

5.4. Bounding the number of NAE satisfying assignments

Here we prove Lemma 5.14, used to prove the key theorem (Theorem 5.8). Recall that $\mathsf{satNAE}(A, \mathbf{C})$ is the string function that outputs the set of clauses in $\mathbf{C}$ that are satisfied as NAE by A (see Section 4). The proof of the following lemma is based on the spectral inequality proved in Section 6.

Lemma 5.14 *(Assuming the premise of the main formula) $\boldsymbol{VTC}^0$ proves*

$$numones(\mathsf{satNAE}(A, \mathbf{C})) \leq (\lambda n + 3m)/4 + \Theta(1/n).$$

Proof. Let $\mathbf{a}$ be a vector from $\{-1, 1\}^n$ such that $\mathbf{a}(i) = 2A(i) - 1$. Thus, $\mathbf{a}(i) = 1$ if $A(i) = 1$ and $\mathbf{a}(i) = -1$ if $A(i) = 0$. We can prove in $\boldsymbol{VTC}^0$ (by definition of inner products and a product of a matrix and a vector—*innerprod* and *Matvecprod* function symbols, respectively, as defined in Section 3.2.2) the following:

$$\mathbf{a}^t M \mathbf{a} = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{ij} \mathbf{a}(i) \mathbf{a}(j). \tag{37}$$

By assumption $\textsc{Mat}(M, \mathbf{C})$ holds (see Definition 4.5) and so by definition 4.5 and by (37) we can prove in $\boldsymbol{VTC}^0$ that:

$$\mathbf{a}^t M\mathbf{a} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=0}^{m-1} E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j), \tag{38}$$

where $E_{ij}^{(k)}$, for any $i, j \in [n]$, is:

$$E_{ij}^{(k)} := \begin{cases} +\frac{1}{2}, & x_i^{\varepsilon_i}, x_j^{\varepsilon_j} \in \mathbf{C}[k] \text{ and } \varepsilon_i \neq \varepsilon_j, \text{ for some } \varepsilon_i, \varepsilon_j \in \{0,1\} \text{ and } i \neq j; \\ -\frac{1}{2}, & x_i^{\varepsilon_i}, x_j^{\varepsilon_j} \in \mathbf{C}[k] \text{ and } \varepsilon_i = \varepsilon_j, \text{ for some } \varepsilon_i, \varepsilon_j \in \{0,1\} \text{ and } i \neq j; \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

By rearranging (38) we get

$$\mathbf{a}^t M\mathbf{a} = \sum_{k=0}^{m-1}\sum_{i=1}^{n}\sum_{j=1}^{n} E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j),$$

and since $E_{ij}^{(k)} = 0$ whenever either $x_i \notin \mathbf{C}[k]$ or $x_j \notin \mathbf{C}[k]$, we get

$$= \sum_{k=1}^{m-1} \sum_{i,j\in\{r\,:\,x_r\in\mathbf{C}[k]\}} E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j),$$

and further, since $E_{ij}^{(k)} = 0$ if $i = j$, and $E_{ij}^{(k)} = E_{ji}^{(k)}$, for any i, j, we have

$$= \sum_{k=0}^{m-1} \sum_{i<j\in\{r\,:\,x_r\in\mathbf{C}[k]\}} 2E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j). \tag{40}$$

Claim 5.20. *The theory $\boldsymbol{VTC}^0$ (in fact already $\mathbf{V}^0$) proves that for any $k = 0, \ldots, m-1$:*

$$\sum_{i<j\in\{r\,:\,x_r\in\mathbf{C}[k]\}} 2E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j) = \begin{cases} +1, & \mathrm{NAE}(\mathbf{C}[k], A); \\ -3, & \neg\mathrm{NAE}(\mathbf{C}[k], A). \end{cases}$$

Proof of claim: For any $i < j \in \{r : x_r \in \mathbf{C}[k]\}$, if $A(i) \neq A(j)$ (which means that $\mathbf{a}(i) \neq \mathbf{a}(j)$) then $\mathbf{a}(i)\mathbf{a}(j) = -1$, and if $A(i) = A(j)$ (which means that $\mathbf{a}(i) = \mathbf{a}(j)$) then $\mathbf{a}(i)\mathbf{a}(j) = 1$. Note also that $x_i^{\varepsilon_i} \neq x_j^{\varepsilon_j}$ under $\mathbf{a}$ means that either x_i, x_j have different polarities $\varepsilon_i \neq \varepsilon_j$ and $\mathbf{a}(i) = \mathbf{a}(j)$ or

x_i, x_j have the same polarities $\varepsilon_i = \varepsilon_j$ and $\mathbf{a}(i) \neq \mathbf{a}(j)$. Similarly, $x_i^{\varepsilon_i} = x_j^{\varepsilon_j}$ under $\mathbf{a}$ means that either x_i, x_j have different polarities $\varepsilon_i \neq \varepsilon_j$ and $\mathbf{a}(i) \neq \mathbf{a}(j)$ or x_i, x_j have the same polarities $\varepsilon_i = \varepsilon_j$ and $\mathbf{a}(i) = \mathbf{a}(j)$. Thus, by (39), for any $i < j \in \{r : x_r \in \mathbf{C}[k]\}$:

$$E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j) = \begin{cases} +\frac{1}{2}, & \text{if } x_i^{\varepsilon_i} \neq x_j^{\varepsilon_j} \text{ under } \mathbf{a}; \\ -\frac{1}{2}, & \text{if } x_i^{\varepsilon_i} = x_j^{\varepsilon_j} \text{ under } \mathbf{a}. \end{cases} \tag{41}$$

Note that if $\mathrm{NAE}(\mathbf{C}[k], A)$ is true then there are exactly two pairs of literals $x_i^{\varepsilon_i}, x_j^{\varepsilon_j}$, $i < j$, for which $x_i^{\varepsilon_i}$ and $x_j^{\varepsilon_j}$ get different values under the assignment $\mathbf{a}$ (if A assigns 1 (i.e., $\top$) to one literal and 0 (i.e., $\bot$) to the other two literals, then two pairs have different values and one pair has the same value; and similarly if A assigns 0 to one literal and 1 to the other two literals). Therefore, if $\mathrm{NAE}(\mathbf{C}[k], A)$ is true then

$$\sum_{i<j\in\{r\,:\,x_r\in\mathbf{C}[k]\}} 2E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j) = 2\left(\frac{1}{2} + \frac{1}{2} - \frac{1}{2}\right) = 1.$$

On the other hand, if $\mathrm{NAE}(\mathbf{C}[k], A)$ is false then all pairs of literals $x_i^{\varepsilon_i}, x_j^{\varepsilon_j}$, $i < j$, get the same value under the assignment A, and so:

$$\sum_{i<j\in\{r\,:\,x_r\in\mathbf{C}[k]\}} 2E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j) = 2\left(-\frac{1}{2} - \frac{1}{2} - \frac{1}{2}\right) = -3.$$

■Claim

Let $Z = \{i < m : \mathrm{NAE}\,(\mathbf{C}[i], A)\}$ (note that $Z = \mathsf{satNAE}(A, \mathbf{C})$), and for any $k = 0, \ldots, m-1$, let $\gamma_k = \sum_{i<j\in\{r\,:\,x_r\in\mathbf{C}[k]\}} 2E_{ij}^{(k)}\mathbf{a}(i)\mathbf{a}(j)$. Then, by Claim 5.20 and Proposition 3.19:

$$\begin{aligned} \sum_{i=0}^{m-1} \gamma_i &= 1 \cdot \mathit{numones}(Z) - 3 \cdot (m - \mathit{numones}(Z)) \\ &= 4 \cdot \mathit{numones}(Z) - 3m \\ &= 4 \cdot \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C})) - 3m. \end{aligned} \tag{42}$$

By (40) we have

$$\sum_{i=0}^{m-1} \gamma_i = \mathbf{a}^t M \mathbf{a}, \tag{43}$$

and by the spectral inequality proved in Lemma 6.7 in the next section, we have:

$$\mathbf{a}^t M \mathbf{a} \leq \lambda n + \Theta(1/n).$$

By (42) we thus get

$$4 \cdot \mathit{numones}(\mathsf{satNAE}(A, \mathbf{C})) - 3m \leq \lambda n + \Theta(1/n),$$

which leads to

$$\mathit{numones}(\mathsf{satNAE}(A, \mathbf{C})) \leq \frac{\lambda n + 3m}{4} + \Theta(1/n).$$

□

6. The spectral bound

In this section we show how to prove inside $\boldsymbol{VTC}^0$ the desired spectral inequality, used in the proof of the key theorem (Theorem 5.8; specifically, it was used in Lemma 5.14 in Section 5.4).

Since the original matrix associated with a 3CNF is a real symmetric matrix, and its eigenvectors and eigenvalues also might be real, and thus cannot be represented in our theory $\boldsymbol{VTC}^0$, we shall need to work with *rational approximations* of real numbers. We will work with polynomially-bounded approximations. Specifically, a real number r in the real interval $[-1, 1]$ is represented with precision $1/n^{\kappa}$, where n is the number of variables in the 3CNF (that is, if $\widetilde{r}$ is the approximation of r, we shall have $|r - \widetilde{r}| \leq 1/n^{\kappa}$). ***For concreteness we fix*** $\kappa = 7$. Recall that when we use iterated addition of many rational numbers we first make sure that all the denominators are identical, and specifically are $n^{2\kappa} = n^{14}$ (see Section 3.1.2, Fact 2).

The idea of proving the spectral bound in $\boldsymbol{VTC}^0$ (Lemma 6.7). Here we explain informally how to proceed to prove the bound $\mathbf{a}^t M \mathbf{a} \leq \lambda n + \Theta(1/n)$, for any $\mathbf{a} \in \{-1, 1\}^n$, in the theory $\boldsymbol{VTC}^0$, assuming that $\textsc{EigValBound}(M, \vec{\lambda}, V)$ (and $\textsc{Mat}(M, \mathbf{C})$) hold. The idea is as follows: in the predicate $\textsc{EigValBound}(M, \vec{\lambda}, V)$ we certify that the rows of a given matrix V are rational approximations of the normalized eigenvector basis of M. Since M is symmetric and real, V will approximate an orthonormal matrix, and V^t will *approximate* V^{-1} (this is where we circumvent the need to prove the correctness of inverting a matrix in the theory $\boldsymbol{VTC}^0$: instead of proving the existence of an inverse matrix, we simply assume that there exists an object which [approximates] the inverse matrix of V). Thus, V^t approximates the matrix of the basis transformation from the standard basis to the eigenvector basis. Note that $\mathbf{a}$ (as a $\{-1, 1\}$ vector) is already almost described in the standard basis. Hence, it will be possible to prove in the theory that $V^t\mathbf{a}$ is the representation of $\mathbf{a}$ in the (approximate) eigenvector basis, i.e., we shall have an equality $\mathbf{a} = \sum_{i=1}^{n} \gamma_i \mathbf{v}_i + o(1)$, for $\mathbf{v}_i$'s the approximate eigenvectors of M and some rationals γ_i's. After plugging in this equality in $\mathbf{a}^t M \mathbf{a}$, to prove $\mathbf{a}^t M \mathbf{a} \le \lambda n$ we only need to *validate computations*—using also the fact that we know that the inequalities $M\mathbf{v}_i \le \lambda \mathbf{v}_i + o(1)$, for any $i \in [n]$, hold (since this will be stated in the predicate $\textsc{EigValBound}(M, \vec{\lambda}, V)$).

6.1. Notations

Here we collect the notation we use in this section. We denote by $e_1, \ldots, e_n$ the standard basis vectors spanning $\mathbb{Q}^n$. That is, for any $1 \le i \le n$ the vector $e_i \in \mathbb{Q}^n$ is 1 in the ith coordinate and all other coordinates are 0. For a vector $\mathbf{v}$ we denote by $\mathbf{v}(j)$ the jth entry in $\mathbf{v}$. Given a real symmetric matrix M we denote by $\mathbf{u}_1, \ldots, \mathbf{u}_n \in \mathbb{R}^n$ the normalized eigenvectors of M. It is known that the collection of normalized eigenvectors of a symmetric $n \times n$ real matrix M forms an orthonormal basis for $\mathbb{R}^n$, called *the eigenvector basis of* M (cf. [30]). The (rational) approximation of the eigenvectors will be denoted $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{Q}^n$ and we define $v_{ij} := \mathbf{v}_i(j)$. Recall that for a real or rational vector $v = (v_1, \ldots, v_n)$ we denote by $\|v\|^2$ the squared Euclidean norm of v, that is, $\|v\|^2 = v_1^2 + \ldots + v_n^2$. We also define $\|\mathbf{v}\|_\infty := \max\{v_i : 1 \le i \le n\}$.

6.2. Rational approximations of real numbers, vectors and matrices

Definition 6.1 (Rational ε-approximation of a real number). *For $r \in \mathbb{R}$, we say that $q \in \mathbb{Q}$ is* a rational ε-approximation of r *(or just* ε-approximation*), if $|r - q| \le \varepsilon$.*

Claim 6.2. *For any real number $r \in [-1, 1]$ and any natural number m there exists a $1/m$-approximation of r whose numerator and denominator have values linearly bounded in m.*

Proof of claim: By assumption, there exists an integer $0 \le k < 2m$, such that $r \in \left[-1 + \frac{k}{m}, -1 + \frac{k+1}{m}\right]$. Then $-1 + \frac{k}{m}$ is a rational $1/m$-approximation of r. $\blacksquare_{\text{Claim}}$

In a similar fashion we have:

Definition 6.3 (Rational ε-approximation of (sets of) real vectors). *Let $0 < \varepsilon < 1$. For $\mathbf{u} \in \mathbb{R}^n$, we say that $\mathbf{v} \in \mathbb{Q}^n$ is* an ε-approximation of $\mathbf{u}$*, if $\mathbf{v}(i)$ is an ε-approximation of $\mathbf{u}(i)$, for all $i = 1, \ldots, n$. Accordingly, for a* *sequence* $U = (\mathbf{u}_1, \ldots, \mathbf{u}_k) \in (\mathbb{R}^n)^k$, *we say that* $V = (\mathbf{v}_1, \ldots, \mathbf{v}_k) \in (\mathbb{Q}^n)^k$ *is a* (rational) ε-approximation of U *if every* $\mathbf{v}_i \in \mathbb{Q}^n$ *is an* ε*-approximation of the vector* $\mathbf{u}_i$, $i = 1, \ldots, n$.

6.3. The predicate EigValBound

We define the predicate EigValBound$(M, \vec{\lambda}, V)$, which is meant to express the properties needed for the main proof. Basically, EigValBound$(M, \vec{\lambda}, V)$ expresses the fact that V is a rational $1/n^{\kappa}$-approximation (Definition 6.3) of the eigenvector basis of M, whose $1/n^{\kappa}$-approximate eigenvalues (in decreasing order with respect to value) are $\vec{\lambda}$ (for $\kappa = 7$) (recall that the denominator of every rational number in the theory is $n^{2\kappa}$).

Note: For a number or a number-term in the language, we sometimes use $|t|$ to denote the absolute value of t. This should not be confused with the length $|T|$ of a string term T.

Definition 6.4 (EigValBound predicate)**.** *The predicate* EigValBound$(M, \vec{\lambda}, V)$ *is a* Σ_0^B*-definable relation in* $\boldsymbol{VTC}^0$ *that holds (in the standard two-sorted model) iff all the following properties hold:*

1. V *is a sequence of* n *vectors* $\mathbf{v}_1, \ldots, \mathbf{v}_n \in \mathbb{Q}^n$ *with polynomially small entries. That is, for any* $1 \le i, j \le n$, *the rational number*

$$v_{ij} := \mathbf{v}_i(j) \in \mathbb{Q}$$

   is polynomial in n *(meaning that both its denominator and numerator are polynomially bounded in* n*).*
2. *For any* $1 \le i, j \le n$ *it holds that the absolute value* $|v_{ij}| \le 2$.
3. *For any* $1 \le i \le n$, *define:*

$$\widetilde{e}_i := \sum_{j=1}^{n} v_{ij} \cdot \mathbf{v}_j\, .$$

   Then, there exists $\mathbf{r}_i \in \mathbb{Q}^n$ *for which*

$$\widetilde{e}_i = e_i + \mathbf{r}_i \quad \textit{and} \quad \|\mathbf{r}_i\|_\infty = O(1/n^{\kappa-1}).$$

   To formalize the existence of such an $\mathbf{r}_i$ *we do not use an existential second-sort quantifier here; instead, we simply assert that for any* $\ell = 1, \ldots, n$:

$$|\widetilde{e}_i(\ell) - e_i(\ell)| = O(1/n^{\kappa-1}).$$

4. *The vectors in V are "almost" orthonormal, in the following sense:*

$$\begin{aligned} \langle \mathbf{v}_i, \mathbf{v}_j \rangle &= O(1/n^{\kappa-1})\,, && \textit{for all } 1 \le i \ne j \le n, \\ \langle \mathbf{v}_i, \mathbf{v}_i \rangle &= 1 + O(1/n^{\kappa-1})\,, && \textit{for all } 1 \le i \le n\,. \end{aligned}$$

5. *The parameter $\vec{\lambda}$ is a sequence $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$ of rational numbers such that for every $1 \le i \le n$, there exists a vector $\mathbf{t}_i \in \mathbb{Q}^n$ for which $\|\mathbf{t}_i\|_\infty = O(1/n^{\kappa-3})$, and*

$$M\mathbf{v}_i = \lambda_i \mathbf{v}_i + \mathbf{t}_i\,.$$

(Similar to Item 3 above, we do not use an existential second-sort quantifier for $\mathbf{t}_i$ here.)

It should be easy to check that $\textsc{EigValBound}(M, \vec{\lambda}, V)$ is a Σ_0^B-definable relation in $\boldsymbol{VTC}^0$.

Now we show that there exist objects $M, \vec{\lambda}, V$ that satisfy the predicate $\textsc{EigValBound}(M, \vec{\lambda}, V)$.

Proposition 6.5 (Suitable approximations of eigenvector bases exist)**.** *Let M be an $n \times n$ real symmetric matrix whose entries are quadratic in n. Let $U = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\} \subseteq \mathbb{R}^n$ be the orthonormal basis consisting of the eigenvectors of M. If $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subseteq \mathbb{Q}^n$ is an $1/n^\kappa$-approximation of U (Definition 6.3), $\vec{\lambda} = \{\lambda_1, \ldots, \lambda_n\}$ is the collection of rational $1/n^\kappa$-approximations of the real eigenvalues of M such that $\lambda_1 \ge \lambda_2 \ge \ldots \ge \lambda_n$, then $\textsc{EigValBound}(M, \vec{\lambda}, V)$ holds (as before, the predicate holds in the standard two-sorted model, for the appropriate* encodings *of its parameters).*[9]

Proof. Let u_{ij} be an abbreviation of $\mathbf{u}_i(j)$, that is, the jth element in the vector $\mathbf{u}_i$, and similarly for v_{ij}. We proceed by checking each of the conditions in Definition 6.4.

Condition (1): Holds by the definition of an approximation of a real vector and by Claim 6.2, stating that the ε-approximation of a real number in $[-1, 1]$ is a rational number whose denominator and numerator both are of value $O(n^\kappa)$.

[9] This is an existence statement. We do not claim that the statement of the proposition is provable in the theory (nevertheless, some of the computations can be carried out inside the theory).

Condition (2): Since v_{ij} is a rational $1/n^{\kappa}$-approximation of u_{ij}, and $|u_{ij}| \le 1$ (because $\|\mathbf{u}_i\| = 1$) for any $1 \le i, j \le n$, we have $|v_{ij}| \le 2$.

Condition (3): We will identify the set U of eigenvectors with its associated matrix consisting of the eigenvectors as columns. By orthonormality of this real matrix U, we have that $U^t = U^{-1}$, that is:

$$\sum_{i=1}^{n} u_{ij}\mathbf{u}_i = e_j \text{ , for any } j = 1, \ldots, n\,. \tag{44}$$

By assumption, for any $1 \le i \le n$ there exists $\mathbf{s}_i = (s_{i1}, \ldots, s_{in}) \in \mathbb{R}^n$ such that $\|\mathbf{s}_i\|_\infty \le 1/n^{\kappa}$ and $\mathbf{v}_i = \mathbf{u}_i + \mathbf{s}_i$. Therefore, for any $1 \le j \le n$, we have:

$$\begin{aligned}\widetilde{e}_j := \sum_{i=1}^{n} v_{ij}\mathbf{v}_i &= \sum_{i=1}^{n} (u_{ij} + s_{ij}) \cdot (\mathbf{u}_i + \mathbf{s}_i) \\ &= \underbrace{\sum_{i=1}^{n} u_{ij}\mathbf{u}_i}_{=e_j \text{ by (44)}} + \sum_{i=1}^{n} u_{ij}\mathbf{s}_i + \sum_{i=1}^{n} s_{ij} \cdot (\mathbf{u}_i + \mathbf{s}_i)\,. \end{aligned} \tag{45}$$

We define

$$\mathbf{r}_j := \sum_{i=1}^{n} u_{ij}\mathbf{s}_i + \sum_{i=1}^{n} s_{ij} \cdot (\mathbf{u}_i + \mathbf{s}_i)\,,$$

which gives us

$$\widetilde{e}_j = e_j + \mathbf{r}_j\,.$$

Note that since $\sum_{i=1}^{n} v_{ij}\mathbf{v}_i = \tilde{e}_j$ is a rational vector then $\mathbf{r}_j$ is also a rational vector.

It remains to show that $\|\mathbf{r}_j\|_\infty = O(1/n^{\kappa-1})$. Since $1 = \|\mathbf{u}_i\|^2 = \sum_{j=1}^{n} u_{ij}^2$, we have $|u_{ij}| \le 1$. By this, and by the fact that $\|\mathbf{s}_i\|_\infty \le 1/n^{\kappa}$, we get $\|\sum_{i=1}^{n} u_{ij}\mathbf{s}_i\|_\infty = O(1/n^{\kappa-1})$, and $\left\|\sum_{i=1}^{n} s_{ij} \cdot (\mathbf{u}_i + \mathbf{s}_i)\right\|_\infty = O(1/n^{\kappa-1})$. This means that $\|\mathbf{r}_j\|_\infty = O(1/n^{\kappa-1})$.

Condition (4): This is similar to the proof of Condition (3). By assumption, for any $1 \le i \le n$ there exists $\mathbf{s}_i = (s_{i1}, \ldots, s_{in}) \in \mathbb{R}^n$ such that $\|\mathbf{s}_i\|_\infty \le 1/n^{\kappa}$, and $\mathbf{v}_i = \mathbf{u}_i + \mathbf{s}_i$. Thus, we have

$$\begin{aligned}\langle \mathbf{v}_i, \mathbf{v}_j \rangle &= \langle \mathbf{u}_i + \mathbf{s}_i, \mathbf{u}_j + \mathbf{s}_j \rangle \\ &= \langle \mathbf{u}_i, \mathbf{u}_j \rangle + \langle \mathbf{s}_i, \mathbf{u}_j + \mathbf{s}_j \rangle + \langle \mathbf{u}_i, \mathbf{s}_j \rangle\,. \end{aligned} \tag{46}$$

The first term in (46) is 0 since U is an orthonormal basis, and the second and third terms in (46) are both $O(1/n^{\kappa-1})$ (by calculations similar to that in the proof of Condition (3)).

The proof of $\langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1 + O(1/n^{\kappa-1})$, for all $1 \le i \le n$, is similar.

Condition (5): Similar to the proof of previous conditions, we define $\mathbf{s}_i = (s_{i1}, \ldots, s_{in}) \in \mathbb{R}^n$ such that $\|\mathbf{s}_i\|_\infty \le 1/n^\kappa$, and $\mathbf{v}_i = \mathbf{u}_i + \mathbf{s}_i$, for any $1 \le i \le n$. We have

$$\begin{aligned} M\mathbf{v}_i &= M(\mathbf{u}_i + \mathbf{s}_i) \\ &= M\mathbf{u}_i + M\mathbf{s}_i. \end{aligned} \tag{47}$$

Since $\mathbf{u}_i \in \mathbb{R}^n$ is the eigenvector of M and λ_i is a $1/n^\kappa$-approximation of the eigenvalue of $\mathbf{u}_i$, we have that (47) equals

$$(\lambda_i + \epsilon)\mathbf{u}_i + M\mathbf{s}_i \tag{48}$$

for some $|\epsilon| \le 1/n^\kappa$,

$$\begin{aligned} &= \lambda_i \mathbf{u}_i + \epsilon \mathbf{u}_i + M\mathbf{s}_i \\ &= \lambda_i(\mathbf{v}_i - \mathbf{s}_i) + \epsilon \mathbf{u}_i + M\mathbf{s}_i \\ &= \lambda_i \mathbf{v}_i - \lambda_i \mathbf{s}_i + \epsilon \mathbf{u}_i + M\mathbf{s}_i\,. \end{aligned}$$

We put

$$\mathbf{t}_i := -\lambda_i \mathbf{s}_i + \epsilon \mathbf{u}_i + M\mathbf{s}_i.$$

It remains to show that $\|\mathbf{t}_i\|_\infty = O(1/n^{\kappa-3})$.

Claim 6.6. *For every* $1 \le i \le n$, $\lambda_i = O(n^3)$.

Proof of claim: Since $\|\mathbf{u}_i\|_\infty = 1$ and, by assumption, every entry in M is $O(n^2)$, we have:

$$\|M\mathbf{u}_i\|_\infty = O(n^3). \tag{49}$$

Observe that

$$M\mathbf{u}_i = (\lambda_i + \epsilon)\mathbf{u}_i = \lambda_i \mathbf{u}_i + \epsilon \mathbf{u}_i. \tag{50}$$

Because $|\epsilon| \le 1/n^\kappa$ and $\|\mathbf{u}_i\|_\infty = 1$, we have $\|\epsilon \mathbf{u}_i\|_\infty = O(1/n^\kappa)$. Therefore, by (49) and (50) we have $\lambda_i = O(n^3)$. $\blacksquare_{\text{Claim}}$

We have $\|\mathbf{s}_i\|_\infty \le 1/n^\kappa$, and so by Claim 6.6 we get that $\| - \lambda_i \mathbf{s}_i\|_\infty = O(1/n^{\kappa-3})$. Now, $\|\epsilon \mathbf{u}_i\|_\infty = O(1/n^\kappa)$ and since M has entries which are $O(n^2)$ we have $\|M\mathbf{s}_i\|_\infty = O(1/n^{\kappa-3})$. We conclude that

$$\begin{aligned} \|\mathbf{t}_i\|_\infty &= \| - \lambda_i \mathbf{s}_i + \epsilon \mathbf{u}_i + M\mathbf{s}_i\|_\infty \\ &\le \|-\lambda_i \mathbf{s}_i\|_\infty + \|\epsilon \mathbf{u}_i\|_\infty + \|M\mathbf{s}_i\|_\infty \\ &= O(1/n^{\kappa-3}). \end{aligned}$$

□

6.4. Certifying the spectral inequality

In this section we show that the theory $\boldsymbol{VTC}^0$ can prove that, if $\textsc{EigValBound}(M, \vec{\lambda}, V)$ holds, then the desired spectral inequality also holds.

Note on coding and formalizing the proof in $\boldsymbol{VTC}^0$: In what follows we will write freely terms such as matrices, vectors, inner products, products of a matrix by a vector (of the appropriate dimensions), addition of vectors, and big sums. We also freely use basic properties of these objects; like transitivity of inequalities, distributivity of a product over big sums, associativity of addition and product, etc. We showed how to formalize these objects and how to prove their basic properties within $\boldsymbol{VTC}^0$ in Sections 3.2.2 and 3.2.4 (see Proposition 3.22).

For an assignment $A \in \{0,1\}^n$ we define its associated vector $\mathbf{a} \in \{-1,1\}^n$ such that $\mathbf{a}(i) = 1$ if $A(i) = 1$ and $\mathbf{a}(i) = -1$ if $A(i) = 0$. In other words, we define $\mathbf{a}(i) = 2A(i) - 1$. Note that

$$\mathbf{a} = \sum_{i=1}^{n} \mathbf{a}(i) \cdot e_i \,.$$

We define

$$\widetilde{\mathbf{a}} := \sum_{i=1}^{n} \mathbf{a}(i) \cdot \widetilde{e_i} \,, \tag{51}$$

and recall that $\widetilde{e_i} := \sum_{j=1}^{n} v_{ij} \cdot \mathbf{v}_j$ is a rational approximation of e_i (Definition 6.4). We let $\mathbf{a}^t M \mathbf{a}$ abbreviate $\langle \mathbf{a}, M\mathbf{a} \rangle$ (which is Σ_1^B-definable in $\boldsymbol{VTC}^0$, by Section 3.2.2).

The following lemma is a reiteration of Lemma 5.5 in Section 5.1:

Lemma 6.7 (Main spectral bound)**.** *The theory $\boldsymbol{VTC}^0$ proves that if A is an assignment to n variables (that is, A is a string variable of length $n+1$) and $\textsc{EigValBound}(M, \vec{\lambda}, V)$ holds, then*

$$\mathbf{a}^t M \mathbf{a} \le \lambda n + \Theta(1/n) \,. \tag{52}$$

This is a corollary of Lemma 6.8 and Lemma 6.11 that follow.

Lemma 6.8. *The theory $\boldsymbol{VTC}^0$ proves that for any assignment A to n variables, $\textsc{EigValBound}(M, \vec{\lambda}, V)$ implies:*

$$\mathbf{a}^t M \mathbf{a} \le \widetilde{\mathbf{a}}^t M \widetilde{\mathbf{a}} + \Theta(1/n^{\kappa - 6}).$$

Proof. First note that A is a string variable of length n. By Definition 6.4, for any $1 \le j \le n$ there exists a vector $\mathbf{r}_j \in \mathbb{Q}^n$ such that $\widetilde{e}_j = e_j + \mathbf{r}_j$, and where $\|\mathbf{r}_j\|_\infty = O(1/n^{\kappa-1})$. Therefore, by (51):

$$\widetilde{\mathbf{a}} = \sum_{i=1}^{n} \mathbf{a}(i)\widetilde{e}_i = \sum_{i=1}^{n} \mathbf{a}(i)(e_i + \mathbf{r}_i) = \sum_{i=1}^{n} \mathbf{a}(i)e_i + \sum_{i=1}^{n} \mathbf{a}(i)\mathbf{r}_i \,.$$

Note that $\sum_{i=1}^{n} \mathbf{a}(i)e_i = \mathbf{a}$, and let

$$\mathbf{r} := \sum_{i=1}^{n} \mathbf{a}(i)\mathbf{r}_i \,.$$

Then,

$$\widetilde{\mathbf{a}} = \mathbf{a} + \mathbf{r} \,,$$

and since $\mathbf{a}(i) \in \{-1, 1\}$, we have $\|\mathbf{r}\|_\infty = O(1/n^{\kappa-2})$. Now, proceed as follows:

$$\begin{aligned} \mathbf{a}^t M \mathbf{a} &= (\widetilde{\mathbf{a}} - \mathbf{r})^t M (\widetilde{\mathbf{a}} - \mathbf{r}) \\ &= \widetilde{\mathbf{a}}^t M \widetilde{\mathbf{a}} - \widetilde{\mathbf{a}}^t M \mathbf{r} - \mathbf{r}^t M \widetilde{\mathbf{a}} + \mathbf{r}^t M \mathbf{r} \,. \end{aligned} \tag{53}$$

We now claim that (provably in $\boldsymbol{VTC}^0$) the addition of the three right terms in (53) is $O(1/n^{\kappa-6})$:

Claim 6.9. *The theory $\boldsymbol{VTC}^0$ proves that for any assignment A to n variables,* $\textsc{EigValBound}(M, \vec{\lambda}, V)$ *implies:*

$$-\widetilde{\mathbf{a}}^t M \mathbf{r} - \mathbf{r}^t M \widetilde{\mathbf{a}} + \mathbf{r}^t M \mathbf{r} = O\left(1/n^{\kappa-6}\right) .$$

Proof of claim: Consider $-\widetilde{\mathbf{a}}^t M \mathbf{r}$. Since $\|\widetilde{\mathbf{a}}\|_\infty \le 2$, and since (by construction) each entry in M is at most $O(n^2)$, we have $\|\widetilde{\mathbf{a}}^t M\|_\infty = O(n^3)$. Therefore, since $\|\mathbf{r}\|_\infty \le 1/n^{\kappa-2}$, we get $-\widetilde{\mathbf{a}}^t M \mathbf{r} = O\left(\frac{1}{n^{\kappa-6}}\right)$. Similarly, we have $-\mathbf{r}^t M \widetilde{\mathbf{a}} = O\left(\frac{1}{n^{\kappa-6}}\right)$.

Considering $\mathbf{r}^t M \mathbf{r}$, we have $\|\mathbf{r}^t M\|_\infty = O(1/n^{\kappa-5})$ and so $\mathbf{r}^t M \mathbf{r} = O(1/n^{\kappa-5} \cdot 1/n^{\kappa-2} \cdot n) = O(1/n^{2\kappa-8}) = O(1/n^{\kappa-6})$. $\blacksquare_{\text{Claim}}$

Claim 6.9 concludes the proof of Lemma 6.8. □

Claim 6.10. *The theory $\boldsymbol{VTC}^0$ proves that* $\textsc{EigValBound}(M, \vec{\lambda}, V)$ *implies that:*

$$\begin{aligned} \langle \widetilde{e}_i, \widetilde{e}_i \rangle &= 1 + O(1/n^5), && \textit{for any } 1 \le i \le n, \textit{ and} \\ \langle \widetilde{e}_i, \widetilde{e}_j \rangle &= O(1/n^5), && \textit{for any } 1 \le i \ne j \le n. \end{aligned}$$

Proof of claim: By assumption for any $1 \le i \le n$, $\widetilde{e}_i = e_i + \mathbf{r}_i$ for some $\|\mathbf{r}_i\|_\infty = O(1/n^{\kappa-1})$. Thus

$$\begin{aligned}
\langle \widetilde{e}_i, \widetilde{e}_i \rangle &= \langle e_i + \mathbf{r}_i, e_i + \mathbf{r}_i \rangle \\
&= \|e_i\|^2 + 2\langle e_i, \mathbf{r}_i \rangle + \|\mathbf{r}_i\|^2 \\
&= 1 + O(1/n^{\kappa-1}) + O(n/n^{\kappa-1}) \\
&= 1 + O(1/n^{\kappa-2}) \\
&= 1 + O(1/n^5),
\end{aligned}$$

Proving $\langle \widetilde{e}_i, \widetilde{e}_j \rangle = O(1/n^5)$ for any $1 \le i \ne j \le n$ is similar (in fact $\langle \widetilde{e}_i, \widetilde{e}_j \rangle = O(1/n^9)$). $\blacksquare_{\text{Claim}}$

Lemma 6.11. *The theory $\boldsymbol{VTC}^0$ proves that for any assignment A to n variables,* $\textsc{EigValBound}(M, \vec{\lambda}, V)$ *implies:*

$$\widetilde{\mathbf{a}}^t M \widetilde{\mathbf{a}} \le \lambda n + \Theta(1/n)\,. \tag{54}$$

Proof. We have:

$$\begin{aligned}
\widetilde{\mathbf{a}}^t M \widetilde{\mathbf{a}} &= \widetilde{\mathbf{a}}^t M \left(\sum_{i=1}^n \mathbf{a}(i) \widetilde{e}_i \right) && \text{(by definition of } \widetilde{\mathbf{a}}\text{)} \\
&= \widetilde{\mathbf{a}}^t M \left(\sum_{i=1}^n \left(\mathbf{a}(i) \cdot \sum_{j=1}^n v_{ji} \mathbf{v}_j \right) \right) && \text{(by definition of } \widetilde{e}_i\text{)} \\
&= \widetilde{\mathbf{a}}^t \sum_{i=1}^n \left(\mathbf{a}(i) \cdot \sum_{j=1}^n v_{ji} M \mathbf{v}_j \right) && \text{(rearranging)} \\
&= \widetilde{\mathbf{a}}^t \sum_{i=1}^n \left(\mathbf{a}(i) \cdot \sum_{j=1}^n v_{ji} (\lambda_j \mathbf{v}_j + \mathbf{r}_j) \right) && \text{(by Definition 6.4)} \\
&= \widetilde{\mathbf{a}}^t \sum_{i=1}^n \left(\mathbf{a}(i) \cdot \sum_{j=1}^n \lambda_j v_{ji} \mathbf{v}_j \right) + \underbrace{\widetilde{\mathbf{a}}^t \sum_{i=1}^n \left(\mathbf{a}(i) \cdot \sum_{j=1}^n v_{ji} \mathbf{r}_j \right)}_{①} && \text{(rearranging)}
\end{aligned} \tag{55}$$

We first bound (inside $\boldsymbol{VTC}^0$) the second term above, denoted ①:

Claim 6.12. *The theory $\boldsymbol{VTC}^0$ proves that for any assignment A to n variables,* EigValBound$(M, \vec{\lambda}, V)$ *implies*

$$\widetilde{\mathbf{a}}^t \sum_{i=1}^{n} \left(\mathbf{a}(i) \cdot \sum_{j=1}^{n} v_{ji} \mathbf{r}_j \right) = O(1/n^{\kappa-6}) \,.$$

Proof of claim: The proof is similar to the proof of Claim 6.9. Specifically, by Definition 6.4, for any $1 \le j \le n$, we have $\|\mathbf{r}_j\|_\infty \le 1/n^{\kappa-1}$, and for any $1 \le i, j \le n$, we have $|v_{ji}| \le 2$. Thus, $\boldsymbol{VTC}^0$ proves that $\| \sum_{j=1}^{n} v_{ji} \mathbf{r}_j \|_\infty = O(1/n^{\kappa-2})$, for any $1 \le i \le n$. Since $\mathbf{a}(i) \in \{-1, 1\}$, for any $1 \le i \le n$, the theory $\boldsymbol{VTC}^0$ proves $\|\mathbf{a}(i) \cdot \sum_{j=1}^{n} v_{ji} \mathbf{r}_j\|_\infty = O(1/n^{\kappa-2})$, for any $1 \le i \le n$, and therefore also proves

$$\left\| \sum_{i=1}^{n} \left(\mathbf{a}(i) \cdot \sum_{j=1}^{n} v_{ji} \mathbf{r}_j \right) \right\|_\infty = O(1/n^{\kappa-3}). \tag{56}$$

Now consider $\widetilde{\mathbf{a}} = \sum_{i=1}^{n} \mathbf{a}(i) \widetilde{e}_i = \sum_{i=1}^{n} \left(\mathbf{a}(i) \cdot \sum_{j=1}^{n} v_{ji} \mathbf{v}_j \right)$. Since, for any $1 \le i, j \le n$ we have $|v_{ji}| \le 2$ we have $\| \sum_{j=1}^{n} v_{ji} \mathbf{v}_j \|_\infty = O(n)$. Thus, since $\mathbf{a}(i) \in \{-1, 1\}$, $\boldsymbol{VTC}^0$ can prove that $\widetilde{\mathbf{a}} = O(n^2)$, and so by (56) the theory can finally prove

$$\widetilde{\mathbf{a}}^t \sum_{i=1}^{n} \left(\mathbf{a}(i) \cdot \sum_{j=1}^{n} v_{ij} \mathbf{r}_j \right) = O(1/n^{\kappa-6}).$$

■Claim

It remains to bound the first term in (55):

$$\widetilde{\mathbf{a}}^t \cdot \left(\sum_{i=1}^{n} \mathbf{a}(i) \sum_{j=1}^{n} \lambda_j v_{ji} \mathbf{v}_j \right) . \tag{57}$$

By the definition of $\widetilde{\mathbf{a}}$ in (51) and the definition of the $\widetilde{e}_i$'s, we get that (57) equals:

$$\left(\sum_{i=1}^{n} \mathbf{a}(i) \sum_{j=1}^{n} v_{ji} \mathbf{v}_j^t \right) \cdot \left(\sum_{i=1}^{n} \mathbf{a}(i) \sum_{j=1}^{n} \lambda_j v_{ji} \mathbf{v}_j \right) . \tag{58}$$

We can prove in $\boldsymbol{VTC}^0$ that for any vectors $\mathbf{b}_1, \ldots, \mathbf{b}_\ell \in \mathbb{Q}^n$ and any rational numbers $c_1, \ldots, c_\ell$ and $\zeta_1, \ldots, \zeta_\ell$, such that $\zeta = \max\{\zeta_i : 1 \le i \le$

$\ell\}$, we have

$$\left\langle \sum_{i=1}^{\ell} c_i \mathbf{b}_i, \sum_{i=1}^{\ell} \zeta_i c_i \mathbf{b}_i \right\rangle \leq \zeta \cdot \left\langle \sum_{i=1}^{\ell} c_i \mathbf{b}_i, \sum_{i=1}^{\ell} c_i \mathbf{b}_i \right\rangle .$$

Therefore, we can prove in $\boldsymbol{VTC}^0$ that (58) is at most:

$$\begin{aligned}
&\lambda \cdot \left(\sum_{i=1}^{n} \mathbf{a}(i) \sum_{j=1}^{n} v_{ji} \mathbf{v}_j^t \right) \cdot \left(\sum_{i=1}^{n} \mathbf{a}(i) \sum_{j=1}^{n} v_{ji} \mathbf{v}_j \right) \\
&= \lambda \cdot \left(\sum_{i=1}^{n} \mathbf{a}(i) \widetilde{e}_i^t \right) \cdot \left(\sum_{i=1}^{n} \mathbf{a}(i) \widetilde{e}_i \right) && \text{(by definition of } \widetilde{e}_i\text{)} \\
&= \lambda \cdot \left\langle \sum_{i=1}^{n} \mathbf{a}(i) \tilde{e}_i \, , \, \sum_{i=1}^{n} \mathbf{a}(i) \tilde{e}_i \right\rangle \\
&= \lambda \cdot \sum_{i=1}^{n} \langle \mathbf{a}(i) \tilde{e}_i \, , \, \mathbf{a}(i) \tilde{e}_i \rangle + \lambda \cdot \sum_{1 \leq i \neq j \leq n}^{n} \langle \mathbf{a}(i) \tilde{e}_i, \mathbf{a}(i) \tilde{e}_j \rangle && \text{(by rearranging)} \\
&= \lambda \cdot \sum_{i=1}^{n} \mathbf{a}(i)^2 \langle \tilde{e}_i \, , \, \tilde{e}_i \rangle + \lambda \cdot \sum_{1 \leq i \neq j \leq n}^{n} \mathbf{a}(i) \mathbf{a}(j) \langle \tilde{e}_i, \tilde{e}_j \rangle && \text{(by rearranging again)} \\
&= \lambda \cdot \sum_{i=1}^{n} 1 \cdot (1 + O(1/n^4)) + \lambda \cdot \sum_{1 \leq i \neq j \leq n}^{n} \mathbf{a}(i) \mathbf{a}(j) O(1/n^4) && \text{(by Claim 6.10)} \\
&= \lambda n + O(\lambda / n^4)
\end{aligned}$$

We have shown before that $\lambda = O(n^3)$, hence the last term is $\lambda n + O(1/n)$, concluding the proof of Lemma 6.11. □

7. Wrapping up the proof: $\boldsymbol{TC}^0$-Frege refutations of random 3CNFs

In this section we establish the main result of this paper, namely, polynomial-size $\boldsymbol{TC}^0$-Frege refutations for random 3CNF formulas with $\Omega(n^{1.4})$ clauses.

7.1. Converting the main formula into a $\forall\Sigma_0^B$ formula

Note that the main formula (Definition 4.6) is a $\Sigma_0^B(\mathcal{L})$ formula, where the language $\mathcal{L}$ contains function symbols not in $\mathcal{L}_A^2$, and in particular it contains the *numones* function. Since Theorem 3.25 relates $\boldsymbol{VTC}^0$ proofs of Σ_0^B formulas to polynomial-size $\boldsymbol{TC}^0$-Frege proofs, in order to use this theorem we need to convert the main formula into a Σ_0^B formula (in the language $\mathcal{L}_A^2$). It suffices to show that $\boldsymbol{VTC}^0$ proves that the main formula is equivalent to a $\forall\Sigma_0^B$ formula, since if $\boldsymbol{VTC}^0$ proves a $\forall\Sigma_0^B$ formula $\forall\Phi$, it also proves the Σ_0^B formula Φ obtained by discarding all the universal quantifiers in $\forall\Phi$.

Lemma 7.1. *The theory $\boldsymbol{VTC}^0$ proves that the main formula is equivalent to a $\forall\Sigma_0^B$ formula $\forall\Phi$ where the universal quantifiers in the front of the formula all quantify over string variables that serve as counting sequences. Specifically,*

$$\forall\Phi := \forall Z_1 \leq t_1 \ldots \forall Z_r \leq t_r \, \Phi(Z_1, \ldots, Z_r), \tag{59}$$

where $t_1, \ldots, t_r$ are number terms and $\Phi(Z_1, \ldots, Z_r)$ has also free variables other than the Z_i's, and every occurrence of every Z_i appears in Φ in the form $(\delta_{\mathsf{NUM}}(|T|, T, Z_i) \wedge Z_i[t] = s)$, for some string term T and number terms t, s, and where $\delta_{\mathsf{NUM}}(|T|, T, Z_i)$ states that Z_i is a counting sequence *that counts the number of ones in T until position $|T|$ (see Definition 3.12).*

Proof. The following steps convert the main formula into a $\forall\Sigma_0^B$ formula which is equivalent (provably in $\boldsymbol{VTC}^0$) to the main formula:

1. All the functions in the main formula are $\mathbf{AC}^0$-reducible to $\mathcal{L}_A^2 \cup \{numones\}$ (see Section 3.2.1). Thus, the defining axioms of all the function symbols in the main formula can be assumed to be $\Sigma_0^B(numones)$ formulas. Now, it is a standard procedure to substitute in the main formula all the function symbols by their $\Sigma_0^B(numones)$-defining axioms.[10] The resulting formula is $\Sigma_0^B(numones)$, and provably in $\boldsymbol{VTC}^0$ is equivalent to the original main formula.
2. We now substitute all the *numones* function symbols by their Σ_1^B-defining axioms. Specifically, every occurrence of $numones(t, T)$ in the formula, for t, T number and string terms, respectively, occurs inside some atomic formula $\Psi := \Psi(\ldots numones(t, T) \ldots)$. And so we substitute Ψ by the existential formula

$$\exists Z \leq 1 + \langle |T|, |T| \rangle \left(\delta_{\mathsf{NUM}}(|T|, T, Z) \wedge Z[t] = z \wedge \Psi(\ldots z \ldots)\right).$$

[10] When the defining axiom of a string function $F(\vec{x}, \vec{X})$ is a *bit-definition* $i < r(\vec{x}, \vec{X}) \wedge \psi(i, \vec{x}, \vec{X})$ we substitute an atomic formula like $F(\vec{x}, \vec{X})(z)$, by $z < r(\vec{x}, \vec{X}) \wedge \psi(z, \vec{x}, \vec{X})$ (cf. Lemma V.4.15. in [17]).

3. Note that all the *numones* function symbols appear in the *premise* of the implication in the main formula, so we can take all these existential quantifiers out of the premise of the implication and obtain a universally quantified formula, where the universal quantifiers in the front of the formula all quantify over string variables that serve as counting sequences (as in Item 2 above).

□

7.2. Propositional proofs

We need to restate the main probabilistic theorem in [22]:

Theorem 7.2 ([22], Theorem 3.1)**.** *Let* $\mathbf{C}$ *be a random 3CNF with* n *variables and* $m = \beta \cdot n$ *clauses, for* $\beta = c \cdot n^{0.4}$*, where* c *a sufficiently large constant. Then, with probability converging to* 1 *as* $n \longrightarrow \infty$*, the following hold:*

- *The imbalance of* $\mathbf{C}$ *is at most* $O(n\sqrt{\beta})$*;*
- *The largest eigenvalue* λ *satisfies* $\lambda = O(\sqrt{\beta})$*;*
- *There are* $k = O(n^{0.2})$*,* $t = \Omega(n\beta)$ *(and* $t < n^2$*),* $d = O(k)$*, and* $\mathcal{C}$ *with* $|\mathcal{C}| = t$*, such that* $\textsc{Coll}(t, k, d, n, m, \mathbf{C}, \mathcal{C})$ *holds.*

We need to rephrase the theorem in a manner that suites our needs, as follows:

Corollary 7.3. *Let* $\mathbf{C}$ *be a random 3CNF with* n *variables and* $m = \beta n$ *clauses, where* $\beta = cn^{0.4}$*, for* c *a sufficiently large constant. Then, with probability converging to* 1 *as* $n \longrightarrow \infty$*, the following hold:*[11]

1. *There exists an* $I = O(n\sqrt{\beta}) = O(n^{1.2})$ *such that* $\textsc{Imb}(\mathbf{C}, I)$*.*
2. *Let* $\kappa = 7$*. There exists a* $1/n^{\kappa}$*-rational approximation* V *of the eigenvector matrix of* M *and* $1/n^{\kappa}$*-rational approximations* $\vec{\lambda}$ *of the eigenvalues of* M*; in other words,* $\textsc{EigValBound}(M, \vec{\lambda}, V)$ *and* $\textsc{Mat}(M, \mathbf{C})$ *hold. And the* $1/n^{\kappa}$*-rational approximation* λ *of the largest eigenvalue of* M *satisfies* $\lambda = O(\sqrt{\beta}) = O(n^{0.2})$*.*

[11] Formally speaking, we mean that the following three items hold in the standard two-sorted model $\mathbb{N}_2$, when all the second-sort objects (like $\mathbf{C}$ and $\mathscr{D}$) are in fact finite sets of numbers (encoding $\mathbf{C}$ and $\mathscr{D}$), natural numbers are treated as natural numbers in the standard two-sorted model and rational numbers are the corresponding natural numbers that encode them as pairs of natural numbers (as described in Section 3.1.2).

3. *There are natural numbers* $k = O(n^{0.2})$, $t = \Omega(n\beta) = \Omega(n^{1.4})$ *(and also* $t < n^2$*),* $d = O(k) = O(n^{0.2})$ *and a sequence* $\mathscr{D}$ *of* t *inconsistent* k*-tuples such that* $\textsc{Coll}(t, k, d, n, m, \mathbf{C}, \mathscr{D})$ *holds, and such that (where* c' *is the sufficiently large constant term from the main formula):*

$$t > \frac{d(I + \lambda n)}{2} + \frac{c'd}{n}. \tag{60}$$

Proof. The corollary stems from Theorem 7.2, when plugging-in the value $cn^{0.4}$ for β. Item 2 follows from Proposition 6.5. For the third item, we first plug-in the value $cn^{0.4}$ for β, and then we need to verify inequality (60). For this, note that if we substitute the parameters in the inequality, we get $\Omega(n\beta)$ in the left hand side, and $O(n^{0.2}(n\sqrt{\beta} + n\sqrt{\beta})/2) = O(n^{1.2}\sqrt{\beta})$ in the right hand side of (60). Thus, (60) holds for *a sufficiently large constant* c (where c is the constant from $\beta = cn^{0.4}$). □

Recall the premise in the implication in the main formula:

$$\text{3CNF}(\mathbf{C}, n, m) \wedge \textsc{Coll}(t, k, d, n, m, \mathbf{C}, \mathscr{D}) \wedge \textsc{Imb}(\mathbf{C}, I) \wedge \textsc{Mat}(M, \mathbf{C}) \wedge \\ \textsc{EigValBound}(M, \vec{\lambda}, V) \wedge \lambda = \max\{\vec{\lambda}\} \wedge t > \frac{d \cdot (I + \lambda n)}{2} + c'd/n. \tag{61}$$

Let $\text{PREM}(\mathbf{C}, n, m, t, k, d, \mathscr{D}, I, \vec{\lambda}, V, M, \lambda, \vec{Z})$ be the formula obtained from (61) after transforming the main formula into a $\forall\Sigma_0^B$ formula, where $\vec{Z}$ is a sequence of strings variables for counting sequences added after the transformation (as described in Lemma 7.1).

The following is a simple claim about the propositional translation (given without a proof):

Claim 7.4. *If a* Σ_0^B *formula* $\varphi(\vec{x}, \vec{X})$ *can be evaluated to a true sentence in* $\mathbb{N}_2$ *by assigning numbers* $\underline{\vec{x}}$ *and sets* $\underline{\vec{X}}$ *to the appropriate variables, then the translation* $[\![\varphi]\!]_{\underline{\vec{x}}, |\underline{\vec{X}}|}$ *is satisfiable.*

Lemma 7.5. *For every* $m, n \in \mathbb{N}$ *and every unsatisfiable 3CNF formula* $\mathbf{C}$ *with* m *clauses and* n *variables such that* $\text{PREM}(\mathbf{C}, n, m, \dots)$ *is true for some assignment to the remaining variables (i.e. to the unspecified variables denoted by "*$\dots$ *"; this also implies that* $[\![\text{PREM}(\mathbf{C}, n, m, \dots)]\!]$ *is satisfiable), there exists a polynomially bounded* $\boldsymbol{TC}^0$*-Frege proof of* $\neg\mathbf{C}$ *(i.e. the sequent* $\longrightarrow \neg\mathbf{C}$ *can be derived).*

Proof. Recall that for given $m, n \in \mathbb{N}$, 3CNF formula $\mathbf{C} = (\mathbf{C}[\alpha])_{\alpha<m}$ and assignment A, the formula $\exists\alpha \leq m\,\textsc{NotSAT}(\mathbf{C}[i], A)$ (which is the consequence of the implication in the main formula 4.6) is the statement:

$$\begin{aligned} \exists\alpha < m\exists i, j, k \leq n\big(\quad & \langle\mathbf{C}[\alpha]\rangle_1^5 = i \land (A(i) \leftrightarrow \langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_1^3 = 0) \\ & \land \langle\mathbf{C}[\alpha]\rangle_2^5 = j \land (A(j) \leftrightarrow \langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_2^3 = 0) \\ & \land \langle\mathbf{C}[\alpha]\rangle_3^5 = k \land (A(k) \leftrightarrow \langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_3^3 = 0)\big). \end{aligned}$$

The propositional translation of this formula (Definition 3.23) contains the variables $p^{\mathbf{C}}_{\langle i,j,k,\ell,\alpha\rangle}$ with $i, j, k \leq n$, $\alpha < m$ and ℓ describes the signs of the literals. Additionally it contains variables p_i^A for $i \leq n$ stemming from the assignment A. It is not necessary to show the full translation of the formula, since we intend to plug-in propositional constants $(\top, \bot)$ for some of the variables. In other words, parts of the formula will consist only of constants and so it is unnecessary to give these parts in full detail. Having this in mind, the translation $\llbracket\exists\alpha < m\textsc{NotSAT}(\mathbf{C}[\alpha], A)\rrbracket_{m,n}$ is

$$\begin{aligned} \bigvee_{\alpha=0}^{m-1}\bigvee_{i,j,k=1}^{n}\Big(\quad & (\llbracket\langle\mathbf{C}[\alpha]\rangle_1^5 = i\rrbracket_{m,n} \land (p_i^A \leftrightarrow \llbracket\langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_1^3 = 0\rrbracket_{m,n})) \\ & \land (\llbracket\langle\mathbf{C}[\alpha]\rangle_2^5 = j\rrbracket_{m,n} \land (p_j^A \leftrightarrow \llbracket\langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_2^3 = 0\rrbracket_{m,n})) \\ & \land (\llbracket\langle\mathbf{C}[\alpha]\rangle_3^5 = k\rrbracket_{m,n} \land (p_k^A \leftrightarrow \llbracket\langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_3^3 = 0\rrbracket_{m,n}))\Big). \end{aligned} \tag{62}$$

Here, the variables $p^{\mathbf{C}}_{\langle i,j,k,\ell,\alpha\rangle}$ all implicitly appear in the parts inside $\llbracket\cdot\rrbracket$.

Now assume we have a fixed 3CNF $\underline{\mathbf{C}}$ with n variables and m clauses. Then for every $\alpha < m$ there exist $1 \leq i, j, k \leq n$ such that the formulas $\llbracket\langle\underline{\mathbf{C}[\alpha]}\rangle_1^5 = i\rrbracket_{m,n}$ and $\llbracket\langle\underline{\mathbf{C}[\alpha]}\rangle_2^5 = j\rrbracket_{m,n}$ and $\llbracket\langle\underline{\mathbf{C}[\alpha]}\rangle_3^5 = k\rrbracket_{m,n}$ are all satisfied (in fact they are polynomial-size in n propositional tautologies consisting of only constants $\top, \bot$). From now on we will only concentrate on the disjuncts where this is the case (as the other disjuncts are falsified, or in other words they are propositional contradictions consisting of only constants).

By plugging $\underline{\mathbf{C}}$ into $\llbracket\langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_1^3 = 0\rrbracket_{m,n}$ and $\llbracket\langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_2^3 = 0\rrbracket_{m,n}$ and $\llbracket\langle\langle\mathbf{C}[\alpha]\rangle_4^5\rangle_3^3 = 0\rrbracket_{m,n}$ we get that $\llbracket\exists\alpha < m\textsc{NotSAT}(\mathbf{C}[\alpha], A)\rrbracket_{m,n}$ is evaluated (modulo a trivial simplification) to

$$\bigvee_{\alpha<m}\Big((p_i^A)^{\ell_1^\alpha} \land (p_j^A)^{\ell_2^\alpha} \land (p_k^A)^{\ell_3^\alpha}\Big), \tag{63}$$

where ℓ_r^α is an abbreviation of $\llbracket\langle\langle\underline{\mathbf{C}[\alpha]}\rangle_4^5\rangle_r^3 = 0\rrbracket_{m,n}$, and thus we can observe that (63) gets evaluated to $\neg\underline{\mathbf{C}}(p_1^A/x_1, \ldots, p_n^A/x_n)$, where p_i^A/x_i means substitution of x_i by p_i^A.

By Theorem 5.8 the theory $\boldsymbol{VTC}^0$ proves the main formula and so by Lemma 7.1 there is a $\boldsymbol{VTC}^0$ proof of

$$\mathrm{PREM}(\mathbf{C}, n, m, t, k, d, \mathscr{D}, I, \vec{\lambda}, V, M, \lambda, \vec{Z}) \to \exists i < m\, \textsc{NotSAT}(\mathbf{C}[i], A).$$

Thus, by Theorem 3.25 we can derive a polynomially bounded $\boldsymbol{TC}^0$-proof of the formula

$$[\![\mathrm{PREM}(\mathbf{C}, \dots)]\!]_{m,n} \to [\![\exists \alpha < m\textsc{NotSAT}(\mathbf{C}[\alpha], A)]\!]_{m,n}$$

and thus also of the *sequent*

$$[\![\mathrm{PREM}(\mathbf{C}, \dots)]\!]_{m,n} \longrightarrow [\![\exists \alpha < m\textsc{NotSAT}(\mathbf{C}[\alpha], A)]\!]_{m,n}.$$

By Claim 7.4 and the assumption that $\mathrm{PREM}(\underline{\mathbf{C}}, n, m, \dots)$ is true in $\mathbb{N}_2$ for an assignment to the remaining variables we know that $[\![\mathrm{PREM}(\underline{\mathbf{C}}, \dots)]\!]_{m,n}$ is satisfiable. Plugging in such a satisfying assignment $\vec{a}$ into $[\![\mathrm{PREM}(\underline{\mathbf{C}}, \dots)]\!]_{m,n}$, Lemma 2.7 yields a polynomially bounded $\boldsymbol{TC}^0$-Frege proof of

$$[\![\mathrm{PREM}(\underline{\mathbf{C}}, \vec{a})]\!]_{m,n}$$

and of the sequent

$$[\![\mathrm{PREM}(\underline{\mathbf{C}}, \vec{a})]\!]_{m,n} \longrightarrow [\![\exists \alpha < m\textsc{NotSAT}(\underline{\mathbf{C}[\alpha]}, A)]\!]_{m,n}.$$

Using the Cut rule (Definition 2.4) we get a polynomially bounded $\boldsymbol{TC}^0$-Frege proof of the formula

$$[\![\exists \alpha < m\textsc{NotSAT}(\underline{\mathbf{C}[\alpha]}, A)]\!]_{m,n}.$$

As we showed before, this gets evaluated to

$$\neg\underline{\mathbf{C}}(p_1^A/x_1, \dots, p_n^A/x_n)$$

as desired. Because of Claim 2.7, this proof is only polynomially longer than the one of the translation of the main formula. Since that proof was polynomially bounded, the above proof of $\neg\underline{\mathbf{C}}(p_i^A/x_i)$ also is. □

We can now conclude:

Corollary 7.6. *With probability converging to* 1*, a random 3CNF* $\mathbf{C}$ *with* n *variables and* $m \geq c \cdot n^{1.4}$ *clauses,* c *a sufficiently large constant,* $\neg\mathbf{C}$ *has polynomially bounded* $\boldsymbol{TC}^0$*-Frege proofs, while* $\mathbf{C}$ *has no sub-exponential size resolution refutations (as long as* $m = O(n^{1.5-\epsilon})$*, for* $0 < \epsilon < 1/2$*).*

Proof. By Corollary 7.3, with probability converging to 1 there exists an assignment of numbers and strings $\vec{\alpha}$ (including also the appropriate counting sequences assigned to the Z_i string variables introduced in Lemma 7.1) such that $\mathrm{PREM}(\mathbf{C}, \vec{\alpha})$ holds (in the standard two-sorted model). Therefore, with probability converging to 1 we can apply Lemma 7.5 to establish that $\neg\mathbf{C}$ has a short $\boldsymbol{TC}^0$-Frege proof. That with probability converging to 1 there are no sub-exponential size resolution refutations of $\mathbf{C}$ follows from [14, 6, 10]. □

Acknowledgments

We wish to thank Jan Krajíček for very helpful discussions concerning the topic of this paper and for commenting on an earlier manuscript, Emil Jeřabek and Neil Thapen for answering many of our questions about theories of weak arithmetic and for many other insightful comments, and the anonymous reviewer who invested a lot of effort in reading the manuscript and providing us with many valuable comments.

References

[1] Miklós Ajtai. The complexity of the pigeonhole principle. In *Proceedings of the IEEE 29th Annual Symposium on Foundations*

[2] Michael Alekhnovich. Lower bounds for k-DNF resolution on random 3-CNFs. In *Proceedings of the Annual ACM Symposium on the Theory of Computing*, pages 251–256, 2005.

[3] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: non-binomial case. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (Las Vegas, NV, 2001)*, pages 190–199. IEEE Computer Soc., Los Alamitos, CA, 2001.

[4] Michael Alekhnovich and Alexander A. Razborov. Resolution is not automatizable unless W[P] is tractable. *SIAM J. Comput.*, 38(4):1347–1363, 2008.

[5] A. Atserias, Maria Luisa Bonet, and J. Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Information and Computation*, 176:152–136, August 2002.

[6] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. The efficiency of resolution and Davis-Putnam procedures. *SIAM J. Comput.*, 31(4):1048–1075 (electronic), 2002.

[7] Eli Ben-Sasson. *Expansion in Proof Complexity.* PhD thesis, Hebrew University, Jerusalem, Israel, September 2001.

[8] Eli Ben-Sasson and Yonatan Bilu. A gap in average proof complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 2002. TR02-003.

[9] Eli Ben-Sasson and Russell Impagliazzo. Random CNF's are hard for the polynomial calculus. *Computational Complexity*, pages 1–19, 2010.

[10] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *J. ACM*, 48(2):149–169, 2001.

[11] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On interpolation and automatization for Frege systems. *SIAM J. Comput.*, 29(6):1939–1967, 2000.

[12] Samuel R. Buss. *Bounded Arithmetic*, volume 3 of *Studies in Proof Theory.* Bibliopolis, 1986.

[13] Samuel R. Buss and Peter Clote. Cutting planes, connectivity, and threshold logic. *Arch. Math. Logic*, 35(1):33–62, 1996.

[14] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. Assoc. Comput. Mach.*, 35(4):759–768, 1988.

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

[16] Stephen Cook. *Theories for Complexity Classes and Their Propositional Translations*, pages 175–227. Complexity of computations and proofs, Jan Krajíček, ed. Quaderni di Matematica, 2005.

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

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

[19] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7), 1962.

[20] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.

[21] Uriel Feige. Refuting smoothed 3CNF formulas. In *Proceedings of the IEEE 48th Annual Symposium on Foundations of Computer Science*, pages 407–417. IEEE Computer Society, 2007.

[22] Uriel Feige, Jeong Han Kim, and Eran Ofek. Witnesses for non-satisfiability of dense random 3CNF formulas. In *Proceedings of the IEEE 47th Annual Symposium on Foundations of Computer Science*, 2006.

[23] Uriel Feige and Eran Ofek. Easily refutable subformulas of large random 3CNF formulas. *Theory of Computing*, 3(1):25–43, 2007.

[24] Joel Friedman, Andreas Goerdt, and Michael Krivelevich. Recognizing more unsatisfiable random k-SAT instances efficiently. *SIAM J. Comput.*, 35(2):408–430, 2005.

[25] Nicola Galesi and Massimo Lauria. Optimality of size-degree trade-offs for polynomial calculus. *ACM Trans. Comput. Log.*, 12(1), 2011.

[26] A. Goerdt and M. Krivelevich. Efficient recognition of random unsatisfiable k-SAT instances by spectral methods. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 294–304, 2001.

[27] Andreas Goerdt and André Lanka. Recognizing more random unsatisfiable 3-SAT instances efficiently. *Electronic Notes in Discrete Mathematics*, 16:21–46, 2003.

[28] P. Hájek and P. Pudlák. *Metamathematics of First-order Arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1993.

[29] Armin Haken. The intractability of resolution. *Theoret. Comput. Sci.*, 39(2-3):297–308, 1985.

[30] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

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

[32] Emil Jeřábek. Dual weak pigeonhole principle, Boolean complexity, and derandomization. *Ann. Pure Appl. Logic*, 129(1-3):1–37, 2004.

[33] Jan Krajíček. Lower bounds to the size of constant-depth propositional proofs. *The Journal of Symbolic Logic*, 59(1):73–86, 1994.

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

[35] Jan Krajíček. On the weak pigeonhole principle. *Fund. Math.*, 170(1-2):123–140, 2001.

[36] Jan Krajíček. A proof complexity generator. In *Proc. from the 13th International Congress of Logic, Methodology and Philosophy of Science (Beijing, August 2007)*, Studies in Logic and the Foundations of Mathematics. King's College Publications, London, 2009.

[37] Alexis Maciel and Toniann Pitassi. On $\mathrm{ACC}^0[p^k]$ Frege proofs. In *Proceedings of the Annual ACM Symposium on the Theory of Computing 1997 (El Paso, TX)*, pages 720–729 (electronic). ACM, New York, 1999.

[38] Sebastian Müller and Iddo Tzameret. Short Propositional Refutations for Dense Random 3CNF Formulas. In *Proceedings of the 27th Annual ACM/IEEE Symposium on Logic In Computer Science* (LICS), pages 501–510. 2012.

[39] Phuong Nguyen. Proving infinitude of prime numbers using binomial coefficients. In *22nd International Workshop on Computer Science Logic, (Bertinoro, Italy)*, volume 5213 of *Lecture Notes in Computer Science*, pages 184–198. Springer, 2008.

[40] Phuong Nguyen and Stephen A. Cook. Theories for TC^0 and other small complexity classes. *Logical Methods in Computer Science*, 2(1), 2006.

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

[42] Nathan Segerlind, Sam Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM J. Comput.*, 33(5):1171–1200 (electronic), 2004.

[43] Michael Soltys and Stephen Cook. The proof complexity of linear algebra. *Ann. Pure Appl. Logic*, 130(1-3):277–323, 2004.

[44] Iddo Tzameret. Sparser random 3-SAT refutation algorithms and the interpolation problem - (extended abstract). In *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP), track A*, pages 1015–1026, 2014.

[45] Domenico Zambella. Notes on polynomially bounded arithmetic. *The Journal of Symbolic Logic*, 61(3):942–966, 1996.