THEORY OF COMPUTING
www.theoryofcomputing.org

A Simplified Proof of the Space Complexity of Random 4-CNF Formulas in Polynomial Calculus

Massimo Lauria* Mladen Mikša Jakob Nordström† Marc Vinyals‡

December 13, 2025

Abstract. We present a simpler, more explicit proof of the optimal space lower bound for refuting random 4-CNF formulas in polynomial calculus shown by [Bonacina and Galesi '15]. We hope that this makes the result more accessible, and perhaps provides useful insights for other open problems concerning polynomial calculus space.

1 Introduction

Proof complexity studies how hard it is to refute unsatisfiable formulas in conjunctive normal form, i.e., the complement of the SAT problem. Clearly, satisfiable formulas have concise witnesses, as per the definition of NP, but it is less clear how to efficiently certify unsatisfiability. This is just another way of asking how coNP is related to NP, and historically, therefore, the main focus in proof complexity has been

*Partially funded by European Research Council grant ERC-2014-CoG 648276 (AUTAR).

†Supported by the Swedish Research Council grant 2016-00782 and the Independent Research Fund Denmark grant 9040-00389B.

‡This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 802020-ERC-HARMONIC.

ACM Classification: F.2.2, F.4.1

AMS Classification: 03F20

Key words and phrases: proof complexity, polynomial calculus, space complexity, random CNF formulas

on proof size. In the late 1990s, however, work began on studying also the space complexity of proofs. The initial motivation for this came from SAT solving, where memory management is a major concern, but space complexity is of course also a fundamental measure in its own right in computational complexity theory, and the study of space in proof complexity has revealed many connections and questions of intrinsic interest.

To give an informal description of the model, suppose that we have a blackboard on which the proof is presented. The size of a proof is then the number of symbols written on the board. But suppose that we can erase from the board, and that we have the input formula in read-only memory and do not need to have it at all times on the board. Then the space of a proof is how large the board needs to be for a self-contained presentation of the proof, where inference rules can only be applied to formulas currently on the board, but where partial results can be erased either when no longer needed or when space is tight (and we are willing to do the extra work of rederiving the same results later when needed again).

In the interest of brevity, we will focus the rest of this introduction no the space measure in proof complexity, referring the reader to, e.g., the book [21] or the survey chapter [10] for more information about proof complexity in general.

1.1 Previous work on proof space complexity

A formal definition of proof space complexity was first given for the resolution proof system in [14], and this definition was generalized to polynomial calculus[1] and other proof systems in [1]. In resolution new clauses implied by any pair of previous clauses can be derived until the empty clause is obtained, showing that the given formula is unsatisfiable. In polynomial calculus clauses are translated to multilinear polynomials over some field in formal variables x and $\bar{x}$ for all literals occurring in the formula, and one can then take linear combinations and multiply by variables to derive contradiction, i. e., by showing that the polynomial 1 lies in the ideal generated by the polynomials corresponding to the clauses of the CNF formula. Additional polynomials $x^2 - x$ and $x + \bar{x} - 1$ enforce that we only consider Boolean assignments and that the meaning of negation is respected. To date, most work on proof space complexity has studied *clause space* in resolution and *monomial space* in polynomial calculus, measuring the maximal number of clauses/monomials needed simultaneously in a self-contained presentation of a proof that a formula is unsatisfiable.[2]

The focus of this paper is polynomial calculus, but to get a sense for how to prove space lower bounds it is instructive to first look at the simpler case of resolution. Almost all lower bounds on resolution clause space go as follows: Suppose that we want to prove that refuting a CNF formula F requires space at least s. Then, writing $\mathbb{C}_t$ to denote the clauses on the blackboard at time t, we can equivalently show that no derivation $\pi = (\mathbb{C}_1 = \emptyset, \mathbb{C}_2, \ldots, \mathbb{C}_\tau)$ in space less than s can derive contradiction. We do so by inductively constructing partial truth value assignments α_t that assign at most $|\mathbb{C}_t|$ variables and that satisfy all the clauses in $\mathbb{C}_t$ (meaning, in particular, that contradiction has not been derived). The case analysis when we go from $\mathbb{C}_t$ to $\mathbb{C}_{t+1}$ and need to build α_{t+1} from α_t, is as follows:

[1] In this introductory overview we do not distinguish between *polynomial calculus (PC)* [13] and the slightly more general version *polynomial calculus resolution (PCR)* [1], the latter being more relevant from the point of view of space complexity.

[2] We mention for completeness that there is also a *total space* measure counting the total number of variables in memory with repetitions, which has been studied in [1, 6, 7, 9], but we do not discuss this measure in the current paper.

1. If $\mathbb{C}_{t+1}$ is obtained from $\mathbb{C}_t$ by inferring a new clause C from clauses on the board, then α_t already satisfies C (by the soundness of the proof system) and we can set $\alpha_{t+1} = \alpha_t$.

2. If $\mathbb{C}_{t+1}$ is obtained from $\mathbb{C}_t$ by erasing clauses from the board, then α_t also satisfies $\mathbb{C}_{t+1}$, but we need to shrink the assignment to at most $|\mathbb{C}_{t+1}|$ variables. This is not a problem, though, since we can just choose from α_t one variable per clause in $\mathbb{C}_{t+1}$.

3. The tricky case is when a new *axiom clause* C from F is copied to the board. Now we need to argue that if α_t does not already satisfy $\mathbb{C}_{t+1}$ but is still small enough, we can find a variable x in C that is not assigned by α_t, and extend α_t to α_{t+1} by setting x appropriately. In general we may not have such unassigned x in C, nonetheless we do when F is a formula with certain properties of expansion and $\mathbb{C}_t$ is small.

In this way optimal lower bounds on clause space were shown in [1, 4, 14] for many formula families commonly studied in proof complexity. With hindsight, all of these bounds turn out to follow from lower bounds on the *width* of refutations, measured as the size of a largest clause in any refutation [2] (see also [16]). A sequence of papers [23, 25, 5] later separated the space and width measures by proving space lower bounds using fairly different techniques based on pebble games [26, 11, 24], that we do not discuss here.

For polynomial calculus the approach above provably does not work, however, as demonstrated in [1]. What fails is step 2: the refutation may contain small configurations that are satisfiable but not by setting just one variable per monomial, contrary to what happens in resolution (consider the polynomial $x_1x_2\cdots x_n - 1$, which only contains two monomials yet only evaluates to zero when all n variables are set to 1). But if we can instead pick *two* new, unassigned variables x, y for each new clause C in step 3, fix the 2-clause $D \subseteq C$ over these variables, and carry out the line of argument above for the set of all assignments that satisfy such clauses D, step 2 can be made to work also for polynomial calculus. (This fact, called the *locality lemma*, is highly nontrivial, although the proof boils down to elementary combinatorics, and it lies at the foundation also of the ensuing polynomial calculus space lower bounds discussed below.) Thus, instead of maintaining satisfying partial assignments α_t, which can be viewed as 1-CNF formulas, we construct satisfiable 2-CNF formulas $\mathbb{D}_t$ that "overapproximate" the set of polynomials $\mathbb{P}_t$ currently on the board in the sense that $\mathbb{D}_t \vDash \mathbb{P}_t$.

There are additional technical complications, however, and the arguments in [1] are carried out in multi-valued logic, which limits the technique to formulas with very large clauses (such as pigeonhole principle formulas). The paper [18] translated the technique to standard 2-valued logic and established space lower bounds for specially tailored 4-CNF formulas, but in order for this to work the implications $\mathbb{D}_t \vDash \mathbb{P}_t$ could be made to hold only for a certain subset of "well-behaved" assignments (and satisfiability of $\mathbb{D}_t$ hence had to be shown with respect to such assignments). Also, both [1] and [18] failed to prove space lower bounds matching the known linear upper bounds (measured in terms of the number of variables of a formula).

In something of a breakthrough result, [8] presented a unified framework for proving polynomial calculus space lower bounds, capturing all the results in [1, 18] and finally establishing optimal, linear lower bounds for randomly sampled 4-CNF formulas. These lower bounds were achieved at the price of further complications in the proofs, however. The main technical contribution of [8] is to define certain abstract combinatorial objects and show that if such objects can be constructed for a family of formulas,

then space lower bounds follow. It appears quite hard to obtain from this combinatorial argument any intuition about what is actually going on in the polynomial calculus derivation. Following this, [6] managed to extend [8] to few more interesting cases, at the expense of additional complications.

1.2 Our contribution

We extract from the machinery developed in [8] an explicit, simple proof of the optimal, linear space lower bound in polynomial calculus for randomly sampled 4-CNF formulas. Given a polynomial calculus derivation $\pi = (\mathbb{P}_0 = \emptyset, \mathbb{P}_1, \ldots, \mathbb{P}_\tau)$ in sublinear space, we show that (asymptotically almost surely over the randomly sampled formula) one can construct a sequence of 2-CNF formulas $(\mathbb{D}_0 = \emptyset, \mathbb{D}_1, \ldots, \mathbb{D}_\tau)$ with the following properties:

P1. Each $\mathbb{D}_t = Q_t \wedge M_t$ consists of two 2-CNF formulas Q_t and M_t where clauses in Q_t do not share any variables, and nor do clauses in M_t; each variable in each clause in M_t is shared with exactly one clause in Q_t; and each clause in Q_t shares exactly one variable with exactly one clause in M_t. The total number of clauses in $\mathbb{D}_t$ is at most six times the number of monomials in $\mathbb{P}_t$.

P2. For each (polynomial encoding of a) clause in F currently present in $\mathbb{P}_t$ there is a subclause in Q_t.

P3. The variables of M_t come from clauses in F that have previously been on the board.

P4. The implication $\mathbb{D}_t \vDash \mathbb{P}_t$ holds over all truth value assignments (well-behaved or not, and in standard 2-valued logic).

It follows immediately from property P1 that each $\mathbb{D}_t$ is satisfiable, and hence so is $\mathbb{P}_t$ by property P4. Perhaps more interestingly, properties P2 and P3 provide some intuition about what polynomials derived in small space must look like. It is a fairly standard fact that a randomly sampled 4-CNF formula has good expansion properties, that is for any small-to-medium-large set of clauses there will be many variables that occur only in a single clause, and property P2 is derived from that fact. What property P3 says is that even as we take linear combinations of the corresponding polynomials and multiply them with other variables, it is not possible to cancel these variables.

The 2-CNF formulas $\mathbb{D}_t$ actually give slightly more precise structural information than this. It follows from the properties above that if a polynomial p with s monomials is derivable in sublinear space from a randomly sampled 4-CNF formula, then we can force p to evaluate to 0 by assigning at most $O(s)$ variables, notwithstanding p possibly having high degree monomials. Thus, in particular, polynomials such as $x_1 x_2 \cdots x_n - 1$ discussed above cannot be derived in small space from a randomly sampled CNF formula (where all of these claims hold asymptotically almost surely). We find this to be quite an intriguing fact.

We want to emphasize that we do not present any additional technical developments of tools in this paper on top of what can be found in [8]. However, by using and simplifying the ideas in [8] for the particular case of random CNF formulas we can obtain a much more transparent proof, which also allows us to derive the kind of structural information about the derived polynomials discussed in the previous paragraph.

1.3 Outline of this paper

After giving the necessary preliminaries in Section 2, we present our simplified proof of the lower bound in [8] in Section 3. We conclude by discussing some open problems in Section 4.

2 Preliminaries

Let us start by quickly reviewing some basics. A *literal* over a Boolean variable x is either the variable x itself or its negation $\bar{x}$. A *k-clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of exactly k literals. A *k-CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of k-clauses. We think of clauses and CNF formulas as sets so that order is irrelevant and there are no repetitions.

Let $\mathbb{F}$ be a field and consider the ring $\mathbb{F}[x,\bar{x},y,\bar{y},\ldots]$ of polynomials, where x and $\bar{x}$ are distinct formal variables intended to encode opposite literals. We identify 0 with true and 1 with false, and write $x^0 = x$ and $x^1 = \bar{x}$, which should not lead to confusion since all polynomials of interest are multilinear. Then we can write a clause $C = \bigvee x_i^b$ as a monomial $\prod x_i^b$ by taking a the product of its literals, and we have that an assignment satisfies a clause iff the corresponding monomial evaluates to zero iff $x_i = b$ for some literal x_i^b in the clause/monomial.

Definition 2.1 ([1, 13])**.** A polynomial calculus resolution (PCR) *configuration* $\mathbb{P}$ is a set of polynomials over $\mathbb{F}[x,\bar{x},y,\bar{y},\ldots]$. A *PCR refutation* of a CNF formula F is a sequence of configurations $\{\mathbb{P}_0,\ldots,\mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = \emptyset$, polynomial 1 is in $\mathbb{P}_\tau$, and for $0 < t \leq \tau$ we obtain the configuration $\mathbb{P}_t$ from $\mathbb{P}_{t-1}$ by applying one of the following steps:

Axiom download $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is either the monomial containing all literals in some clause $C \in F$, the *Boolean axiom* $x^2 - x$ for some variable x, or the *complementarity axiom* $x + \bar{x} - 1$ for a pair of opposite literals x and $\bar{x}$.

Inference $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is the result of one of the *inference rules*:

- *Linear combination.* $\frac{q \quad r}{\alpha q + \beta r}$
- *Multiplication.* $\frac{q}{x^b q}$

where q, r are polynomials in $\mathbb{P}_{t-1}$, coefficients α, β are elements of the underlying field $\mathbb{F}$, x is a variable of F and $b \in \{0,1\}$.

Erasure $\mathbb{P}_t = \mathbb{P}_{t-1} \setminus \{p\}$, where p is a polynomial in $\mathbb{P}_{t-1}$.

The *monomial space* of a configuration $\mathbb{P}$ is the number of monomials it contains (counted with repetitions) and is denoted $Sp(\mathbb{P})$. The monomial space $Sp(\pi)$ of a refutation π is the maximum monomial space of any configuration in it. Taking the minimum space over all PCR refutations of a formula F, we obtain the monomial space complexity $Sp(F \vdash \bot)$ of refuting F in PCR. When clear from the context we refer to monomial space by calling it just space. This is without ambiguity since in this paper we do not discuss any other notion of space with respect to PCR configurations.

We remark that any unsatisfiable CNF formula of n variables has a refutation in space $O(n)$. This is indeed true for clause space in resolution by [14], and PCR simulates resolution efficiently with respect to space. The main result of this paper shows that this upper bound is asymptotically tight for random CNF formulas, sampled according to the following distribution.

Definition 2.2. A *random k-CNF* formula F with n variables and density Δ is an outcome of drawing Δn clauses with repetition uniformly at random from all k-clauses with n variables. We write $F \sim \mathcal{F}_k^{n,\Delta}$ to denote that F is a formula sampled from this distribution.

A way to understand the hardness of refuting a CNF formula F is to study the expansion of the *clause-variable incidence graph*. The latter is the bipartite graph $G(U \dot\cup V, E)$, where we identify U with the set of clauses and V with the set of variables, and there is an edge (C,x) if variable x occurs in C.

Definition 2.3 (Bipartite expander)**.** In a bipartite graph $G(U \dot\cup V, E)$ the neighbourhood of a set of vertices $A \subseteq U$ is the set of all vertices in V that have an adjacent vertex in A, and is denoted as $N(A)$. We say that $G = (U \dot\cup V, E)$ is a (δ, s)-bipartite expander if for all $A \subseteq U$ with $|A| \leq s$ it holds that $|N(A)| \geq \delta|A|$.

We note for the record that the clause-variable incidence graph of a random formula is an excellent expander almost surely.

Theorem 2.4 ([12])**.** *For every fixed $k \geq 4$ and $\Delta > 0$, there exists $\delta > 0$ such that the graph of a random k-CNF formula sampled from $\mathcal{F}_k^{n,\Delta}$ is a $(2+\delta, \Omega(n))$-bipartite expander with probability $1 - o(1)$.*

3 PCR space of refuting random k-CNF formulas

We now present our alternative exposition of the result in [8] that randomly sampled 4-CNF formulas require maximal, linear monomial space to be refuted in PCR. This is a consequence of the more general statement that the space complexity of refuting a CNF formula is related to the matching properties of its clause-variable incidence graph. Our exposition simplifies the presentation and highlights the role of online-matchings.

As discussed in Section 1.2, the axiom part Q of a shadow configuration is formed by variable-disjoint subclauses of some axioms, which amounts to matching the corresponding axioms to a pair of variables. Since the set of matched axioms changes over time, we need a notion of adaptive matching.

This notion is captured by the following game played by two players on a bipartite graph (i. e. the clause-variable incidence graph of some formula). Player P (she) has some tokens, and at each round she can place one of them on some left vertex (i. e. a clause). Every time she does that, Player D (he) must match that token with a right vertex in its neighborhood (i. e. a variable in that clause) that is not already matched to any other token. Player P can also remove one or more tokens at any round, freeing the corresponding right vertices matched to them.

At some point player D may be unable to match a token, because there is no suitable right vertex available. On a clause-variable incidence graph, for example, player P may place on a set of clauses more tokens than the number of variables occurring in them. To make the game more interesting let us say that at any time of this game there can be at most s tokens placed in total, and at most r tokens on any left vertex. Under these constraints it is interesting to ask whether Player P can trap player D into a corner or,

otherwise, if there is a strategy for player D to play this game indefinitely. This strategy is indeed the notion of *online matching*.

Definition 3.1 (Online matching)**.** Let $G = (U \dot\cup V, E)$ be a bipartite graph. A set of edges $L \subseteq E$ is an r-matching if every vertex in U is incident to at most r edges in L and every vertex in V is incident to at most 1 edge in L. An r-matching is a perfect r-matching from $A \subseteq U$ into $B \subseteq V$ if every vertex in A is incident to exactly r edges in L and no vertex in $V \setminus B$ is incident to L.

A collection of r-matchings $\mathcal{L}$ is an (s,r)-online matching if $\emptyset \in \mathcal{L}$; if $\mathcal{L}$ is closed under taking subsets, in the sense that for any $L \in \mathcal{L}$ and $L' \subseteq L$ we have that $L' \in \mathcal{L}$; and if for each r-matching $L \in \mathcal{L}$ such that $|L| < s$, and for each $v \in U$ incident to less than r edges in L, there exists $w \in V$ such that $L \cup \{(v,w)\} \in \mathcal{L}$.

This notion is also known as a depth one wide-sense nonblocking limited generalized concentrator [15] and, for $r = 1$, as the matching game [4]. It replaces the double matching property that was used in the original proof of the space lower bound [8].

Now we can state our main theorem, which connects the presence of online matchings in the incidence graph of a formula with its hardness with respect to PCR space complexity.

Theorem 3.2. *If the clause-variable incidence graph of a CNF formula F has an $(s,2)$-online matching, then PCR requires monomial space greater than $s/8$ to refute F.*

On the outset, for large s the requirement in Theorem 3.2 seems very strong. It turns out that bipartite graphs with good expansion have online matchings [15].

Theorem 3.3 ([15, Proposition 1])**.** *If G is an $(r+\delta, s)$-bipartite expander, then it has a $(\delta s/2, r)$-online matching.*

The clause-variable incidence graph of a random k-CNF has good expansion by Theorem 2.4, and therefore has linear size online matchings by Theorem 3.3. The result in [8] that random formulas have large space lower bounds is then an immediate corollary of Theorem 3.2. Moreover, we also get the minor improvement that our bound does not depend on k.

Corollary 3.4 ([8, Theorem 5.5])**.** *For every fixed $k \geq 4$ and $\Delta > 0$, a random k-CNF formula sampled from $\mathcal{F}_k^{n,\Delta}$ requires $\Omega(n)$ monomial space to be refuted in PCR almost surely.*

We spend the rest of the section proving Theorem 3.2. The plan is to define, for each configuration in the PCR refutation, a simpler and more structured auxiliary configuration that implies the corresponding one in the refutation. If all configurations in the PCR refutation are small, then all such auxiliary configurations will be small and satisfiable. That contradicts the fact that the last auxiliary configuration implies the last (and therefore unsatisfiable) configuration of the PCR refutation.

Definition 3.5. A *shadow configuration* is a 2-CNF formula $\mathbb{D} = Q \wedge M$ satisfying the following properties. Clauses in Q are over pairwise disjoint sets of variables, and the same applies to clauses in M. Furthermore, each variable in each clause in M is shared with exactly one clause in Q, and each clause in Q shares exactly one variable with exactly one clause in M.

The *axiom part* Q keeps track of which axioms we downloaded: every clause in Q is a subclause of some axiom in F. The *derived part* M approximates concepts that have been derived from axioms that might no longer be in memory. Observe that, by definition, $|Q| = 2|M|$. As an example, the formula $Q \wedge M$ with with $Q = \{\overline{a} \vee x, b \vee y, c \vee z, d \vee \overline{w}\}$ and $M = \{a \vee b, \overline{c} \vee d\}$ is a shadow configuration.

We build the shadow configurations by forward induction. For each $\mathbb{P}_t$ in the refutation we build a new $\mathbb{D}_t$ based on the structure of $\mathbb{D}_{t-1}$ and of the derivation step between $\mathbb{P}_{t-1}$ and $\mathbb{P}_t$. Informally, on axiom download we find a clause with two literals that is variable disjoint with $\mathbb{D}$ and we add it to Q. On inference we do nothing. After an erasure step the shadow configuration still implies the configuration in the proof, but it may now be too large. Therefore we compress it into a smaller shadow configuration by means of the following locality lemma.

Lemma 3.6 (Locality lemma)**.** *Let $\mathbb{D} = Q \wedge M$ be a shadow configuration that implies a configuration $\mathbb{P}$. Then there exists a shadow configuration $\mathbb{D}' = Q' \wedge M'$ that implies $\mathbb{P}$ such that $|M'| \leq 2Sp(\mathbb{P})$ and $Q' \subseteq Q$.*

We prove this crucial lemma at the end of the section and continue with the proof of the main theorem.

Proof of Theorem 3.2. Denote as G the clause-variable incidence graph of F and fix one of its $(s,2)$-online matchings. Let $\pi = (\mathbb{P}_0, \ldots, \mathbb{P}_\tau)$ be a derivation from F in space $Sp(\pi) \leq s/4$. We show by induction over the length of π that every configuration $\mathbb{P}_i$ of π is satisfiable. This is sufficient to prove such derivation is not a refutation.

To do that we build a sequence of shadow configurations $(\mathbb{D}_0, \ldots, \mathbb{D}_\tau)$, so that each $\mathbb{D}_t = Q_t \wedge M_t$:

(a) is satisfiable;

(b) implies $\mathbb{P}_t$; and

(c) respects the inequality $|M_t| \leq 2Sp(\mathbb{P}_t)$.

Alongside a shadow configuration $\mathbb{D}_t$ we maintain a subset of axioms $A_t \subseteq F$ such that every clause in Q_t is a subclause of an axiom in A_t, and a perfect 2-matching L_t from A_t into $Vars(Q_t)$; in fact for a subclause $x^a \vee y^b$ of an axiom $m = x^a y^b m'$ we have $L(m) = \{x, y\}$. The key element needed to maintain the invariants of the sequence is that the proof has small space. Notice that invariant (a) always holds, since we can first satisfy all clauses in M independently (they are defined on pairwise disjoint variables), and then we have at least one unassigned variable in each clause of Q, not shared with any other clause of Q.

Base case For the base case $\mathbb{P}_1 = \emptyset$ it is enough to pick $\mathbb{D}_1 = \emptyset$, $A_1 = \emptyset$, and $L_1 = \emptyset$.

Axiom download We have that $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$ where p is the monomial that encodes an axiom of F. If $\mathbb{D}_{t-1} \vDash p$, it is enough to pick $\mathbb{D}_t = \mathbb{D}_{t-1}$ as in the inference case below. The latter case in particular takes care of the download of boolean and complementary axioms.

Otherwise we need to enlarge the matching and the shadow configuration. Since $\mathbb{D}_{t-1} \nvDash p$, the axiom clause encoded by p is not a superset of any clause in Q_{t-1} and therefore such clause is not in A_{t-1} either. By invariant (c), we have that $|L_{t-1}| = 2|A_{t-1}| = 2|Q_{t-1}| = 4|M_{t-1}| \leq 8Sp(\mathbb{P}_{t-1}) \leq 8Sp(\mathbb{P}_t) - 8 \leq s - 8$.

Therefore by definition of online matching we can extend L_{t-1} to a 2-matching $L' \supset L_{t-1}$ that includes p and two variables $\{x,y\}$ in p not occurring in Q_{t-1}, and therefore, by definition, not occurring in $\mathbb{D}_{t-1}$. Observe that $\mathbb{D}_{t-1} \wedge p$ is satisfiable.

A brief look at the definition reveals that a shadow configuration requires an even sized axiom part. To ensure this we simulate a "dummy" axiom download. Let q be another axiom such that $\mathbb{D}_{t-1} \wedge p \not\vDash q$. Such an axiom must exists otherwise F would be satisfiable and the theorem would hold vacuously. Extending the matching again, which we can do because $|L'| \leq s-6$, there are two new variables $\{z,w\}$ in q independent of $\mathbb{D}_{t-1} \cup \{x,y\}$ and a 2-matching $L_t \supset L'$ from $A_{t-1} \cup \{p,q\}$ into $Vars(Q_{t-1}) \cup \{x,y,z,w\}$).

Let $p' = x^a \vee y^b \subseteq p$ and $q' = z^c \vee w^b \subset q$ be the subclauses of p and q supported over variables $\{x,y\}$ and $\{z,w\}$ respectively. Set $Q_t = Q_{t-1} \cup \{p',q'\}$ and $M_t = M_{t-1} \cup \{x^a \vee z^c\}$.

The construction of $\mathbb{D}_t$ is complete[3] and all the required invariants hold by construction.

Inference It is enough to pick $\mathbb{D}_t = \mathbb{D}_{t-1}$ because $\mathbb{P}_{t-1} \vDash \mathbb{P}_t$ and the space of $\mathbb{P}_t$ increases with respect to the space of $\mathbb{P}_{t-1}$.

Erasure Observe that $\mathbb{P}_{t-1} \vDash \mathbb{P}_t$ holds, just as in the inference case, but the space of $\mathbb{P}_t$ is smaller than the space of $\mathbb{P}_{t-1}$. Therefore $\mathbb{D}_{t-1} = Q_{t-1} \wedge M_{t-1}$ could be too large to be used as the shadow configuration at step t. We need to apply Lemma 3.6 in order to obtain a shadow configuration $\mathbb{D}_t = Q_t \wedge M_t$ that implies $\mathbb{P}_t$ and such that $|M_t| \leq 2Sp(\mathbb{P}_t)$. Since $Q_t \subseteq Q_{t-1}$ we can also take $A_t \subseteq A_{t-1}$ and $L_t \subseteq L_{t-1}$. □

The last and most delicate part of the argument is the locality lemma, for which we will need Hall's marriage theorem.

Theorem 3.7 (Hall's marriage theorem)**.** *Let $G = (U \dot\cup V, E)$ be a bipartite graph. There is a perfect matching from U to V if and only if for all $U' \subseteq U$ it holds that $|N(U')| \geq |U|$.*

To establish the locality lemma, we use a straightforward corollary Hall's marriage theorem as stated next.

Lemma 3.8. *Let $G = (U \dot\cup V, E)$ be a bipartite graph. There is a perfect 2-matching from U to V if and only if for all $U' \subseteq U$ it holds that $|N(U')| \geq 2|U|$.*

Proof. If a 2-matching exists, we can partition V into two parts, each containing a matching, and apply Hall's theorem to each part. Conversely, if the neighbourhood condition holds, we can duplicate each vertex in U, apply Hall's theorem to obtain a matching, and merge back the duplicate vertices into a 2-matching. □

Proof of Lemma 3.6. We begin our construction of $\mathbb{D}'$ by choosing which parts of $\mathbb{D}$ to keep.

Towards this goal we build a bipartite graph $H(U \dot\cup V, E)$ identifying U with the set of distinct monomials in $\mathbb{P}$ and V with the derived part M. Recall that every clause $C \in M$ unambiguously identifies

[3]The exact choice of variables and even polarities for $x^a \vee z^c$ does not matter. At this point the two axioms p and q are implied by Q_t and $x^a \vee z^c$ is redundant, but further down in the refutation the monomial space of some configuration $\mathbb{P}$ may shrink, together with the corresponding shadow configuration (via Lemma 3.6). Then either p' or q' or both may be dropped and $x^a \vee z^c$ may be substituted with some clause that is actually necessary to enforce $\mathbb{P}$.

two clauses in Q that share exactly one variable with C, which we denote $Q(C)$. We add an edge (m,C) to H if monomial m mentions any variable in any of the two clauses $Q(C)$.

We split U into two parts. Let U_1 be a maximal set such that $|N(U_1)| < 2|U_1|$, and let us denote $M_1 = N(U_1)$, and denote as $Q_1 = \bigcup_{C \in M_1} Q(C)$ the $2|M_1|$ clauses in Q with variables in common with M_1. Since $|M_1|$ is small relative to U_1, we can keep it as is.

For the remaining part $U_2 = U \setminus U_1$ and $M_2 = M \setminus M_1$ we observe that any set $A \subseteq U_2$ must satisfy $|N(A) \cap M_2| \geq 2|A|$, otherwise A could be added to U_1, which in turn would not be maximal. Hence, by Lemma 3.8, there is a perfect 2-matching from U_2 into M_2. In particular, any monomial of degree 0 (the empty monomial 1) or 1 is part of U_1.

To follow the rest of the construction it may be helpful to use Figure 1 as a reference. Fix a monomial in $m \in U_2$. The 2-matching associates to m two clauses C_m and D_m in M_2. Each of them corresponds to two clauses in Q, let us respectively call them C'_m, C''_m, D'_m, D''_m. By construction the monomial m has some variable x in common with either C'_m or C''_m, and another variable y in common with either D'_m or D''_m. That is, we can write the monomial m as $x^a y^b m'$ for some $a,b \in \{0,1\}$.

We define a new clause $X_m = x^a \vee y^b$. By construction this clause has exactly one variable in common with exactly one among clauses C'_m or C''_m, and the same for clauses D'_m and D''_m. In other words, $Q(X_m)$ contains one of C'_m or C''_m, and one of D'_m and D''_m.

We set $M'_2 = \bigwedge_{m \in U_2} X_m$, and $Q'_2 = \bigcup_{X_m \in U_2} Q(X_m)$ and we define the new shadow configuration $\mathbb{D}'$ as $M' \wedge Q'$ where $M' = M_1 \wedge M'_2$ and $Q' = Q_1 \wedge Q'_2$. Essentially we start with $\mathbb{D}$ and we substitute M_2 and Q_2 with M'_2 and Q'_2, respectively, to obtain a possibly smaller shadow configuration $\mathbb{D}'$ that we claim satisfies the conclusion of the lemma. Indeed it is immediate to check that $Q' \subseteq Q$ and that

$$|M'| = |M_1| + |M'_2| < 2|U_1| + |U_2| \leq 2Sp(\mathbb{P}).$$

It only remains to verify that $\mathbb{D}' \vDash \mathbb{P}$, and to prove this we argue indirectly. For every assignment α that satisfies $\mathbb{D}'$, we build an assignment β that satisfies $\mathbb{D}$ and agrees with α over every monomial in $\mathbb{P}$. Since β satisfies $\mathbb{P}$ by hypothesis, so does α.

Fix any assignment α and let us build β. We begin setting some variables in $Vars(\mathbb{D}) \setminus Vars(\mathbb{D}') = Vars(Q_2 \setminus Q'_2)$ in a way that satisfies $\mathbb{D} \setminus \mathbb{D}'$ as follows. Pick any clause $C = x^g \vee z^h \in M \setminus M' \subseteq M_2$, and let $\{C' = x^c \vee x'^d, C'' = z^e \vee z'^f\} = Q(C)$ be the two clauses in Q_2 that have one variable each in common with C. Notice that neither clause has any variable in common with any monomial in U_1. We have two cases, depending on whether C was matched to some $m \in U_2$.

If edge (m,C) was in the 2-matching, then $|Vars(C) \cap Vars(X_m)| = 1$, and exactly one of C' and C'' is in Q'. Let us say it is the former without loss of generality.[4] Then we can fix $\beta(z) = h$ in order to satisfy C and fix $\beta(z') = f$ in order to satisfy C''. Clause C' is part of Q' and we do not need to satisfy it yet. If C was not matched then all four variables in C' an C'' are free to set and we can fix $\beta(x) = g$, $\beta(x') = d$, and $\beta(z') = f$ in order to satisfy C, C', and C''.

We fix all remaining unset variables in β (in particular the shaded area of Figure 1) to agree with α. Since these contain $Vars(\mathbb{D}')$ this is enough to satisfy $\mathbb{D}'$, and since by construction β satisfies $\mathbb{D} \setminus \mathbb{D}'$ it follows that β satisfies $\mathbb{D}$ and, by hypothesis, $\mathbb{P}$.

[4] To follow the argument better the reader may refer to Figure 1, and take C_m, C'_m, C''_m to be examples for clauses C, C' and C'' as they are mentioned in the proof.

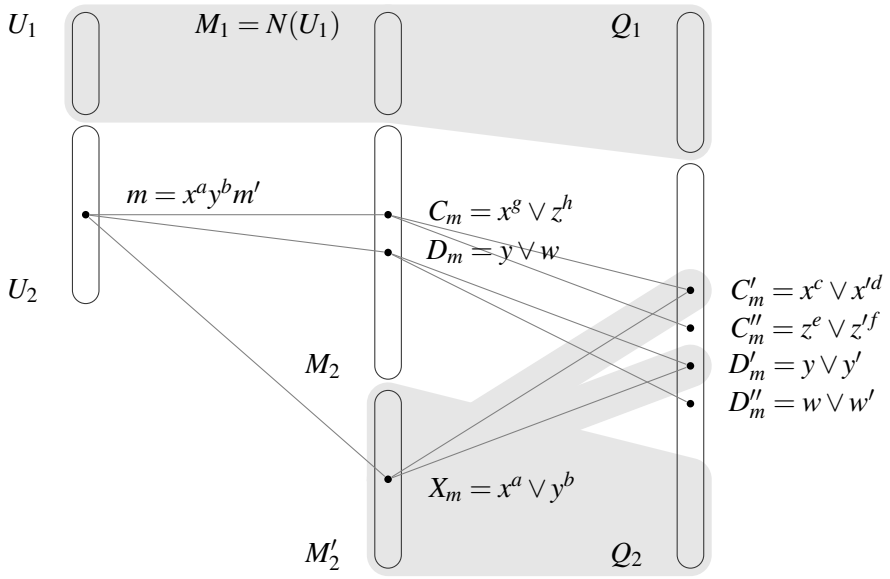

Figure 1: Building a smaller shadow configuration.

We now claim that α satisfies $\mathbb{P}$ as well. On the one hand, since $Vars(U_1) \cap Vars(Q_2) = \emptyset$, monomials in U_1 have all variables assigned to the same value (not necessarily 0) by α and β. On the other hand, even though α and β do not necessarily agree over $Vars(U_2)$, they both satisfy $\mathbb{D}'$, and in particular M'_2, which implies that every monomial $m \in U_2$ evaluates to zero under both assignments. Since both assignments give the same evaluation to the set polynomials $\mathbb{P}$ and β satisfies $\mathbb{P}$, α does too. This concludes our proof that $\mathbb{D}' \vDash \mathbb{P}$. □

4 Concluding remarks

In this paper we present an alternative proof of the result in [8] that refuting random 4-CNF formulas requires linear space in polynomial calculus asymptotically almost surely. Our exposition is simpler in that we avoid the powerful, but somewhat abstract, combinatorial machinery in [8] and instead give an explicit construction of a sequence of 2-CNF formulas that imply the corresponding configurations in a polynomial calculus derivation. These 2-CNF formulas have a nice, intuitive interpretation in that they consist partly of subclauses of axioms currently in memory and partly of clauses tracing derivations made from axioms previously in memory.

We want to emphasize that the general framework in [8] is more powerful than the technique presented in this paper in that it can be used not only to prove lower bound for random CNF formulas but also to unify all previously shown lower bounds for polynomial calculus space. We hope and believe, however, that our way of presenting the lower bound for random CNF formulas, which is arguably the strongest result in [8], can make it accessible to a wider audience.

We also hope that a deeper understanding of the polynomial space lower bound for random CNF formulas can make it possible to attack other open problems concerning space in polynomial calculus.

In contrast to many other lower bound techniques in proof complexity, the polynomial calculus space lower bounds are quite sensitive to transformations such as converting a formula to 3-CNF using auxiliary variables. Thus, while tight lower bounds on space are known for the standard encoding of pigeonhole principle formulas, the only known lower bounds on polynomial calculus space for the canonical 3-CNF version of these formulas are those that follow from the resolution width lower bounds together with [19], which incurs a square root loss. For somewhat related technical reasons there are also no tight lower bounds for functional pigeonhole principle formulas, not even in the standard encoding with wide axioms. Also, although for so-called Tseitin formulas (encoding inconsistent systems of linear equations modulo 2) optimal space lower bounds have been shown for certain subclasses of graphs [17], the problem of establishing that any Tseitin formula over a d-regular expander graph requires linear space remains wide open. For small d, the only space lower bound known for polynomial calculus is the square root of the number of variables, which again follows by appealing to [19]. Quite recently, [3] proved that for large enough d_0, a Tseitin formula over an expander graphs of degree d_0 and n vertices requires $\Omega(n/\log n)$ space, which is almost tight.

Determining the space complexity in polynomial calculus also remains open for so-called ordering principle formulas as studied in, e.g., [20, 22], where the only space lower bounds that can be obtained are worse by a square root than what is known for resolution. And for pebbling formulas nothing no polynomial calculus space lower bounds are known. Pebbling formulas are of particular interest since they exhibit extremal behaviour with respect to the space and width measures in resolution, having refutations in constant width but potentially requiring almost worst-case linear space except for a logarithmic factor [5]—and it would be interesting to determine whether they have the same properties for space versus degree in polynomial calculus.

This leads us to a final, very intriguing question. For resolution it is known that the width complexity of k-CNF formulas is a lower bound on the space complexity [2]. It is natural to ask whether the analogous result holds also for polynomial calculus, whether degree is a lower bound on monomial space. It was shown in [17] that a lower bound on polynomial calculus space in terms of resolution width (which is a weaker measure than polynomial calculus degree) holds for certain types of substituted formulas. More recently, it was established that polynomial calculus space is at least the square root of the resolution width [19], as already alluded to above, but the question of whether this square root loss can be eliminated remains wide open.

Acknowledgements

We are indebted to Ilario Bonacina and Nicola Galesi for numerous and very useful discussions. We would also like to thank Bruno Bauwens, Eli Ben-Sasson, and Yuval Filmus for helpful conversations. Finally, we thank the anonymous referees for helping us improve the presentation.

A Proof of Theorem 3.3

For completeness we reproduce the proof of Theorem 3.3. Our proof goes along the lines of [15], only with minor changes in terminology.

Theorem A.1 (Theorem 3.3, restated)**.** *If G is a $(2s, r+\delta)$-bipartite-expander, then it has an $(s\delta, r)$-online matching.*

To construct an online matching $\mathcal{L}$ we make a few additional definitions. Given an r-matching L, we say a vertex $v \in V$ is idle if it is not incident to any edge in L. Now we define several concepts relative to a set of left-vertices $X \subseteq U$, and to an r-matching L:

- $N_L^{idle}(X)$ is the set of idle vertices that are neighbours of some vertex in X;
- the *assets* of X, denoted as $Ass_L(X)$, is $|N_L^{idle}(X)|$, the number of idle neighbours of X;
- the *liabilities* of X, denoted as $Lia_L(X)$, is $r|X| - |\{(u,v) \in L : u \in X\}|$;
- the *balance* of X, denoted as $Bal_L(X)$, is $Ass_L(X) - Lia_L(X)$;
- we say that X is *solvent* if $Bal_L(X) \geq 0$, *critical* if $Bal_L(X) \leq 0$, and *bankrupt* if $Bal_L(X) < 0$.

We define $\mathcal{L} = \{L : \forall X \subseteq U,\ |X| \leq 2s \to Bal_L(X) \geq 0\}$ as the set of r-matchings L such that every set $X \subseteq U$ of size at most $2s$ is solvent with respect to L. In order to prove that $\mathcal{L}$ is an online matching we need the following properties of the balance function.

Claim A.2. *If $|L| < \delta s$, X is critical, and $|X| \leq 2s$, then $|X| < s$.*

Proof. We have $Ass_L(X) = |N_L^{idle}(X)| \geq |N(X)| - |L| > (r+\delta)|X| - s\delta$, where we bound $|N(X)| \geq (r+\delta)|X|$ by expansion, and $|L| < s\delta$ by hypothesis. Since $Lia_L(X) \leq r|X|$ always holds, we conclude that $0 \geq Bal_L(X) > \delta(|X| - s)$, from where it follows that $|X| < s$. □

Claim A.3. *$Bal_L(X)$ is submodular in X.*

Proof. On the one hand, $Ass_L(X)$ is submodular. This follows from the standard facts $N(X \cup Y) = N(X) \cup N(Y)$, $N(X \cap Y) \subseteq N(X) \cap N(Y)$, and $|A \cup B| + |A \cap B| = |A| + |B|$. Indeed,

$$Ass_L(X \cup Y) + Ass_L(X \cap Y) = |N_i(X \cup Y)| + |N_i(X \cap Y)| \leq |N_i(X) \cup N_i(Y)| + |N_i(X) \cap N_i(Y)| \quad \text{(A.1)}$$

$$= |N_i(X)| + |N_i(Y)| = Ass_L(X) + Ass_L(Y) \quad \text{(A.2)}$$

On the other hand, $Lia_L(X)$ is modular. This is because we can view $Lia_L(X)$ as the modular extension of the element-wise measure $Lia_L(v) = Lia_L(\{v\})$. □

Claim A.4. *If X and Y are critical, $|X|, |Y| \leq 2s$, and $L \in \mathcal{L}$, then $X \cup Y$ is critical.*

Proof. We have

$$Bal_L(X \cup Y) \leq Bal_L(X) + Bal_L(Y) - Bal_L(X \cap Y) \leq -Bal_L(X \cap Y) \leq 0 \quad \text{(A.3)}$$

where the first inequality holds by Claim A.3, the second inequality holds by hypothesis, and the third inequality holds because $X \cap Y$, having size at most $2s$, is solvent with respect to L as prescribed by the definition of $\mathcal{L}$. □

Claim A.5. *Let* $(v,w) \in L$ *and* $K = L \setminus \{(v,w)\}$. *Then* $Bal_K(X) = Bal_L(X) + [\![w \in N(X) \wedge v \notin X]\!]$.

Proof. We have $Ass_K(X) = Ass_L(X) + [\![w \in N(X)]\!]$ because w is idle with respect to K, and $Lia_K(X) = Lia_L(X) + [\![v \in X]\!]$, hence $Bal_K(X) = Bal_L(X) + [\![w \in N(X)]\!] - [\![v \in X]\!]$. Since $v \in X$ implies $w \in N(X)$ a straightforward case analysis shows that we can rewrite the expression as $Bal_K(X) = Bal_L(X) + [\![w \in N(X) \wedge v \notin X]\!]$. □

It remains to prove that $\mathcal{L}$ is indeed an $(s\delta, r)$-online matching. First we show that $\emptyset \in \mathcal{L}$. Fix a set X of size at most $2s$. By expansion and because all vertices are idle $A(X) \geq r|X|$, while $B(X) = r|X|$ because the matching is empty, hence X is solvent.

It follows immediately from Claim A.5 that $\mathcal{L}$ is closed under taking subsets, since for any matchings $L \in \mathcal{L}$, $K = L \setminus \{(v,w)\}$, and set $X \subseteq U$ we have $Bal_K(X) \geq Bal_L(X) \geq 0$.

Finally we show how to extend a matching L given $v \in U$. Assume for the sake of contradiction that for every idle $w \in N_L^{idle}(v)$, $L \cup \{(v,w)\} \notin \mathcal{L}$, and let X_w be a bankrupt set witnessing that. Since X_w is solvent with respect to L and, by Claim A.5, $Bal_L(X)$ and $C_{L\cup(v,w)}(X)$ differ by at most one, we have that X_w is in fact critical with respect to L, that $w \in N(X_w)$, and that $v \notin X_w$.

Let $X = \bigcup_{w \in N_L^{idle}(v)} X_w$. Applying Claims A.2 and A.4 repeatedly, we have that X is critical and $|X| < s$. Let $X' = X \cup \{v\}$. On the one hand, as we just argued, $v \notin X$, which implies that $Lia_L(X') = Lia_L(X) + Lia_L(\{v\}) > Lia_L(X)$. On the other hand $N_L^{idle}(v) \subseteq N_L^{idle}(X)$, which implies that $Ass_L(X') = Ass_L(X)$. We have found a set X' of size at most s that is bankrupt according to L, contradicting that $L \in \mathcal{L}$.

This concludes the proof of Theorem 3.3.

References

[1] MICHAEL ALEKHNOVICH, ELI BEN-SASSON, ALEXANDER A. RAZBOROV, AND AVI WIGDERSON: Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version in *STOC '00*. 2, 3, 5

[2] ALBERT ATSERIAS AND VÍCTOR DALMAU: A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, May 2008. Preliminary version in *CCC '03*. 3, 12

[3] PER AUSTRIN AND KILIAN RISSE: Perfect matching in random graphs is as hard as Tseitin. *TheoretiCS*, 1:22:1–22:47, December 2022. Preliminary version in *SODA '22*. 12

[4] ELI BEN-SASSON AND NICOLA GALESI: Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003. Preliminary version in *CCC '01*. 3, 7

[5] ELI BEN-SASSON AND JAKOB NORDSTRÖM: Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pp. 709–718, October 2008. 3, 12

[6] PATRICK BENNETT, ILARIO BONACINA, NICOLA GALESI, TONY HUYNH, MIKE MOLLOY, AND PAUL WOLLAN: Space proof complexity for random 3-CNFs. *Information and Computation*, 255(1):165–176, August 2017. 2, 4

[7] ILARIO BONACINA: Total space in resolution is at least width squared. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP '16)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 56:1–56:13, July 2016. 2

[8] ILARIO BONACINA AND NICOLA GALESI: A framework for space complexity in algebraic proof systems. *Journal of the ACM*, 62(3):23:1–23:20, June 2015. Preliminary version in *ITCS '13*. 3, 4, 5, 6, 7, 11

[9] ILARIO BONACINA, NICOLA GALESI, AND NEIL THAPEN: Total space in resolution. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS '14)*, pp. 641–650, October 2014. 2

[10] SAMUEL R. BUSS AND JAKOB NORDSTRÖM: Proof complexity and SAT solving. In ARMIN BIERE, MARIJN J. H. HEULE, HANS VAN MAAREN, AND TOBY WALSH, editors, *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, chapter 7, pp. 233–350. IOS Press, 2nd edition, February 2021. Available at http://www.jakobnordstrom.se/publications/. 2

[11] SIU MAN CHAN, MASSIMO LAURIA, JAKOB NORDSTRÖM, AND MARC VINYALS: Hardness of approximation in PSPACE and separation results for pebble games (Extended abstract). In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS '15)*, pp. 466–485, October 2015. 3

[12] VAŠEK CHVÁTAL AND ENDRE SZEMERÉDI: Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988. 6

[13] 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 Theory of Computing (STOC '96)*, pp. 174–183, May 1996. 2, 5

[14] JUAN LUIS ESTEBAN AND JACOBO TORÁN: Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Preliminary versions of these results appeared in *STACS '99* and *CSL '99*. 2, 3, 6

[15] PAUL FELDMAN, JOEL FRIEDMAN, AND NICHOLAS PIPPENGER: Wide-sense nonblocking networks. *SIAM Journal of Discrete Mathematics*, 1(2):158–173, May 1988. 7, 12

[16] YUVAL FILMUS, MASSIMO LAURIA, MLADEN MIKŠA, JAKOB NORDSTRÖM, AND MARC VINYALS: From small space to small width in resolution. *ACM Transactions on Computational Logic*, 16(4):28:1–28:15, July 2015. Preliminary version in *STACS '14*. 3

[17] YUVAL FILMUS, MASSIMO LAURIA, MLADEN MIKŠA, JAKOB NORDSTRÖM, AND MARC VINYALS: Towards an understanding of polynomial calculus: New separations and lower bounds (Extended abstract). *Theory of Computing*, 21(4):1–48, August 2025. Preliminary version in *ICALP '13*. 12

[18] YUVAL FILMUS, MASSIMO LAURIA, JAKOB NORDSTRÖM, NOGA RON-ZEWI, AND NEIL THAPEN: Space complexity in polynomial calculus. *SIAM Journal on Computing*, 44(4):1119–1153, August 2015. Preliminary version in *CCC '12*. 3

[19] NICOLA GALESI, LESZEK KOŁODZIEJCZYK, AND NEIL THAPEN: Polynomial calculus space and resolution width. In *Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS '19)*, pp. 1325–1337, November 2019. 12

[20] NICOLA GALESI AND MASSIMO LAURIA: Optimality of size-degree trade-offs for polynomial calculus. *ACM Transactions on Computational Logic*, 12(1):4:1–4:22, November 2010. 12

[21] JAN KRAJÍČEK: *Proof Complexity*. Volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, March 2019. 2

[22] MLADEN MIKŠA AND JAKOB NORDSTRÖM: A generalized method for proving polynomial calculus degree lower bounds. *Journal of the ACM*, 71(6):37:1–37:43, November 2024. Preliminary version in *CCC '15*. 12

[23] JAKOB NORDSTRÖM: Narrow proofs may be spacious: Separating space and width in resolution. *SIAM Journal on Computing*, 39(1):59–121, May 2009. Preliminary version in *STOC '06*. 3

[24] JAKOB NORDSTRÖM: Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9(3):15:1–15:63, September 2013. 3

[25] JAKOB NORDSTRÖM AND JOHAN HÅSTAD: Towards an optimal separation of space and length in resolution. *Theory of Computing*, 9:471–557, May 2013. Preliminary version in *STOC '08*. 3

[26] MICHAEL S. PATERSON AND CARL E. HEWITT: Comparative schematology. In *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*, pp. 119–127, 1970. 3

AUTHORS

Massimo Lauria
Associate Professor
Department of Statistical Science
Sapienza - Università di Roma
Rome, Italy
massimo.lauria@uniroma1.it
http://www.massimolauria.net/

Mladen Mikša
mladen.miksa@gmail.com

Jakob Nordström
Associate Professor
Department of Computer Science
University of Copenhagen
Copenhagen, Denmark and
Professor
Department of Computer Science
Lund University
Lund, Sweden
jakob.nordstrom@cs.lth.se
http://www.csc.kth.se/~jakobn/

Marc Vinyals
Visiting Fellow
Computer Science Department
Technion
Haifa, Israel
marcviny@cs.technion.ac.il
http://www.csc.kth.se/~vinyals/

ABOUT THE AUTHORS

MASSIMO LAURIA got his PhD at the Department of Computer Science of University "La Sapienza" in Rome, advised by Nicola Galesi. After that he bounced around Europe (and even further) for postdocs and visiting positions, in particular Prague, Stockholm and Barcelona, where he honed his research skill in complexity theory and proof complexity. After that he went back to Rome to join the Department of Statistical Science at University "La Sapienza" in Rome, where he is currently an associate professor.

MLADEN MIKŠA likes to keep an air of mystery about him.

JAKOB NORDSTRÖM got his PhD in 2008 at KTH in Stockholm, advised by Johan Håstad, and after that was a postdoc at the Computer Science and Artificial Intelligence Laboratory at the Massachusetts Institute of Technology until 2010. He worked at KTH as an assistant professor and then associate professor during the years 2011-2019 and now he is a professor at the Department of Computer Science at the University of Copenhagen, Denmark, and also have a part-time affiliation with the the Department of Computer Science at Lund University.

MARC VINYALS did his PhD studies at KTH in Stockholm, advised by Jakob Nordström, and after visiting TIFR in Mumbai he is currently at the Technion in Haifa. His main research interests are proof complexity, communication complexity, and the theory of satisifability solving. When he is not in front of a whiteboard, he can be found near a stage or on top of a kayak.