KTH Computer Science
and Communication

On Complexity Measures in Polynomial Calculus

MLADEN MIKŠA

Doctoral Thesis
Stockholm, Sweden 2016

TRITA-CSC-A 2017:02
ISSN-1653-5723
ISRN-KTH/CSC/A--17/02--SE
ISBN 978-91-7729-226-5

Skolan för datavetenskap och kommunikation
Kungliga Tekniska högskolan
SE-100 44 Stockholm
SVERIGE / SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi fredagen den 20 januari 2017 klockan 14.00 i sal D2, Kungliga Tekniska högskolan, Lindstedtsvägen 3, Stockholm.

Tryck: Universitetsservice US-AB

Abstract

Proof complexity is the study of non-deterministic computational models, called proof systems, for proving that a given formula of propositional logic is unsatisfiable. As one of the subfields of computational complexity theory, the main questions of study revolve around the amount of resources needed to prove the unsatisfiability of various formulas in different proof systems. This line of inquiry has ties to some of the fundamental questions in theoretical computer science, as showing superpolynomial lower bounds on proof size for an arbitrary proof system would separate P from NP. However, while this was the original motivation for the field, that goal of separating P and NP still remains far out of our reach.

In this thesis, we study two simple proof systems: resolution and polynomial calculus. In resolution we reason using clauses, while in polynomial calculus we can use polynomials over some fixed field. We have two main measures of complexity of proofs: size and space. Formally, size is the number of clauses or monomials that appear in a resolution or polynomial calculus proof, respectively. Space is the maximum number of clauses/monomials we need to keep at each time step if we view the proof as being presented as a sequence of configurations of limited memory. A third measure, which turns out to be very important in understanding the others, is width/degree. Width is the size of the largest clause in a resolution proof, while degree is an analogous measure for polynomial calculus that measures the size of a largest monomial in a proof.

One reason that width is important in resolution is that width is a lower bound for space. The original proof of this claim focused on proving a characterization of resolution width in finite model theory and using this characterization to prove the relation with space. In this thesis we give a direct proof of the space-width relation, thereby improving our understanding of it. In the case of polynomial calculus we can pose the question whether the same relation holds between space and degree. We make some progress on this front by showing that if a formula F requires resolution width w then the XORified version of F requires space $\Omega(w)$. On the other hand we show that space lower bounds do not imply degree lower bounds in polynomial calculus, which was already known in resolution.

The second reason why width/degree is an important measure is that strong lower bounds for width/degree imply strong lower bounds for size in both resolution and polynomial calculus. By now, proving width lower bounds in resolution follows a standard process with a developed machinery behind it. However, the situation in polynomial calculus was quite different and degree was much more poorly understood. We improve this situation by providing a unified framework for almost all previous degree lower bounds. Using this framework we also prove a few new degree and size lower bounds. In addition, we explore the relation between theory and practice by running experiments on some current state-of-the-art SAT solvers that are based on resolution.

Sammanfattning

Beviskomplexitet är studiet av icke-deterministiska beräkningsmodeller, så kallade bevissystem, för att bevisa att givna formler i satslogik är osatisfierbara. Som ett delområde inom beräkningsvetenskapen så kretsar de centrala frågorna kring mängden resurser som behövs för att bevisa att givna formler är osatisfierbara i olika bevissystem. Sådana frågeställningar anknyter till fundamentella frågor inom teoretisk datalogi, eftersom superpolynomiella undre gränser på bevisstorlek för ett godtyckliga bevissystem skulle separera P från NP. Även om detta samband var den ursprungliga motivationen till beviskomplexitet, är målet att separera P från NP fortfarande långt bort.

Vi studerar i denna avhandling två enkla bevissystem: resolution och polynomkalkyl. I resolution resonerar man med hjälp av klausuler medan man i polynomkalkyl använder polynom över någon fix kropp. Det finns två huvudsakliga resurser för bevis: storlek och utrymme. Formellt är storlek antalet klausuler som förekommer i ett resolutionsbevis eller antalet monom som förekommer i ett polynomkalkylbevis. Utrymme är definierat som det maximala antalet klausuler eller monom vi måste ha vid något tidssteg om vi betraktar bevis som en sekvens av konfigurationer med begränsat minne. En tredje resurs—som är användbar för att förstå storlek och utrymme—är bredd/gradtal. Bredd definieras som storleken på den största klausulen i ett resolutionsbevis medan gradtal är en motsvarande resurs för polynomkalkyl som mäter storleken på det största monomet i ett bevis.

En anledning till att bredd är en relevant resurs för att förstå resolution är att bredden är en undre gräns för utrymme. Det ursprungliga beviset för detta påstående fokuserade på att bevisa en karakterisering av bredd inom ändlig modellteori och använde denna karakterisering för att bevisa relationen till utrymme. I denna avhandling presenterars ett direkt bevis för utrymme-bredds relationen och tillför därmed till vår förståelse av relationen. För polynomkalkyl kan man fråga om samma relation håller mellan utrymme och gradtal. Vi tillför till denna fråga genom att visa att om en formel F kräver resolutionsbredd w så kräver dess XOR-ifierade version utrymme $\Omega(w)$. Däremot visar vi att undre gränser för utrymme inte innebär undre gränser för gradtal i polynomkalkyl, som tidigare var känt för resolution.

Den andra anledningen till att bredd/gradtal är en relevant resurs är att starka undre gränser för bredd/gradtal innebär starka undre gränser för storlek för både resolution och polynomkalkyl. Vid det här laget följer bevis av undre gränser för bredd i resolution en standardiserad process med sofistikerade matematiska tekniker. Motsvarande process fanns dock inte för polynomkalkyl där gradtal är mycket sämre förstådda. Vi förbättrar denna situation genom att presentera ett enhetligt ramverk för nästan samtliga tidigare undre gränser av gradtal. Med hjälp av detta ramverk visar vi också nya undre gränser för gradtal och storlek. Slutligen undersökar vi relationen mellan teori och praktik genom experiment med några av de främsta moderna SAT lösare som är baserade på resolution.

Acknowledgements

I would like to thank my supervisor Jakob Nordström for introducing me to the field of proof complexity and suggesting research directions, some of which turned into papers that are presented in this thesis. Discussions with Jakob helped me improve my ideas and generate new ones for solving research problems. I have also learned from Jakob the importance of good writing and presentation of ideas in spreading your research.

I would also like to thank Marc Vinyals and Massimo Lauria, who were there from the beginning of my PhD. These were fun years thanks to you two. Since that initial meeting at the Frankfurt airport on our way to interview for PhD positions, Marc and I have shared many discussions on research and other topics. Thank you Marc for them and keep being funny. Massimo provided many interesting views on proof complexity, as well as other non-research related topics. Thank you Massimo for the fun and keep dancing (because who else will).

I would also like to thank current and past members of the proof complexity group: Ilario Bonacina, Susanna de Rezende, Jan Elffers, Jesús Giráldez Crú, and Christoph Berkholz, for interesting research discussions and a pleasant working environment. Out of many research visitors to the proof complexity group, I would especially like to thank Yuval Filmus and Li-Yang Tan.

Thanks to all the previous and current PhD students with whom I shared fikas, some lunches, hiking trips, and dinners. To name them at the peril of forgetting some I thank Adam Schill Collberg, Pedro de Carvalho Gomes, Benjamin Greschbach, Sangxia Huang, Andreas Lindner, Hamed Nemati, Lukáš Poláček, Guillermo Rodríguez Cano, Thatchaphol Saranurak, Freyr Sævarsson, Oliver Schwarz (Oliver is great!), Siavash Soleimanifard, Joseph Swernofsky, Cenny Wenner, and Xin Zhao.

Thank you to my co-advisors Per Austrin and Johan Håstad. Although we did not meet that often, I still learned from them. Also, thank you Per and Cenny for suggesting improvements to the Swedish abstract of my thesis. Thank you to Stefan Arnborg for reading the thesis and suggesting some helpful changes.

Finally, I would like to thank my parents and grandparents for the support and thank you to all my friends for the experiences that we shared.

Contents

Contents **vii**

I Prologue **1**

1 Introduction **3**

2 Background **7**
2.1 Resolution 8
2.2 Polynomial Calculus 10
2.3 Contributions of the Thesis 13

3 Length and Width in Resolution **17**

4 Paper A. Towards an Understanding of Polynomial Calculus **23**
4.1 Space and Degree in Polynomial Calculus 23
4.2 Other Results 24

5 Paper B. From Small Space to Small Width in Resolution **25**
5.1 The New Proof of Space-Width Relation in Resolution 25

6 Paper C. Long Proofs of (Seemingly) Simple Formulas **27**
6.1 Theoretical Hardness of Subset Cardinality Formulas 27
6.2 Experimental Results 28

7 Paper D. A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds **31**
7.1 A Generalized Clause-Variable Incidence Graph 31
7.2 Pigeonhole Principle Bounds 33

8 Conclusion **35**

Bibliography **37**

II Publications **43**

A Towards an Understanding of Polynomial Calculus: New Separations and Lower Bounds **47**

B From Small Space to Small Width in Resolution **95**

C Long Proofs of (Seemingly) Simple Formulas **117**

D A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds **147**

Part I

Prologue

Chapter 1

Introduction

If you recall a time when you tried solving some hard problem, you might recall spending several hours, or even days, in trying to find a solution. However, once you finally knew how to solve it, it likely seemed much simpler and you could check that it was a correct solution with great ease. On the other hand, if the problem did not have any solutions, convincing you of that might have been even harder. Understanding these differences between solving a problem, verifying its solution and establishing that there are no solutions is one of the central fields of interest in computational complexity theory. In this thesis we concentrate specifically on the question of showing that a problem does not have any solutions, which is the main topic of proof complexity.

As an example, let us look at an instance of a sudoku puzzle. Consider an instance in Figure 1.1 and try to solve it. How long did it take you? Most likely more than a couple of minutes. Now, consider if you were given the solution to

					4	7		
						6		
		7	5	6			8	9
		1	2					3
		8				5		
6					7	2		
8	1			5	3	4		
		6						
		9	7					

Figure 1.1: An example of sudoku puzzle.

9	**6**	**3**	**1**	**8**	4	7	**2**	**5**
2	**8**	**5**	**3**	**7**	**9**	6	**1**	**4**
1	**4**	7	5	6	**2**	**3**	8	9
7	**9**	1	2	**4**	**5**	**8**	**6**	3
3	**2**	8	**6**	**9**	**1**	5	**4**	**7**
6	**5**	**4**	**8**	**3**	7	2	**9**	**1**
8	1	**2**	**9**	5	3	4	**7**	**6**
5	**7**	6	**4**	**1**	**8**	**9**	**3**	**2**
4	**3**	9	7	**2**	**6**	**1**	**5**	**8**

Figure 1.2: The solution to the sudoku puzzle in Figure 1.1.

this puzzle as displayed in Figure 1.2. How long does it take you to check that this solution is not a cheat? Likely less than a minute. In general, it seems to us that verifying solutions to problems is much easier than actually solving them. This observation is the intuition behind the main open problem in computational complexity theory, the P vs. NP problem.

We can view the problem of solving a sudoku puzzle in another way as well. Usually, when we are given a sudoku puzzle, we assume that there exists a solution and our "only" task is to find the said solution. However, what would happen if we were given a sudoku puzzle in which we did not know whether a solution existed. How could we prove the existence of a solution, for instance in the example of a puzzle in Figure 1.1? Here the proof would be simple. We would just present the solution from Figure 1.2 and we would be done. If a solution exists the simplest proof that a puzzle is solvable is presenting that solution. Observe that such a proof where we just present the solution is easy to verify and that using this kind of proof cannot establish solvability of an unsolvable puzzle. In other words, the proof makes intuitive sense. One final thing to note in this case is also that this kind of proof is short. That is, our solution is not significantly larger than the specification of the puzzle itself.

Let us look now at a second example of a sudoku puzzle presented in Figure 1.3. Can you solve this puzzle? The first thing that we can notice when trying to solve it is that we get stuck very quickly. After we reach the configuration in Figure 1.4 we are left with no more forced decisions. That is, in order to proceed in solving this puzzle we need to guess the value of one of the squares. However, we run into a problem if we try to do that in this case. No matter what value we choose for our guess, we cannot find a full solution. This puzzle is actually unsolvable! The question we can ask then is how can we convince anyone else of this conclusion? How can we prove to someone that this puzzle is unsolvable?

7		4						5
9	1				7	6	8	
			1					
	3	7		2			5	
			3		9	4	6	
	4			8		3		
4		2			6			
			8		2		4	
1		9				5	2	6

Figure 1.3: A second example of a sudoku puzzle.

7		4	**9**		**8**			5
9	1		**2**		7	6	8	
			1					
	3	7		2			5	
			3		9	4	6	
	4			8		3		
4	**7**	2	**5**		6			
			8		2		4	
1	**8**	9				5	2	6

Figure 1.4: Partially solved sudoku puzzle from Figure 1.3.

One way would be to list all possible guesses and show that all of them lead to an inconsistency in the puzzle. However, if after the first guess we at some point need to guess again, the number of choices we need to list in our proof doubles. Thus, following this strategy for proving the unsolvability of the puzzle could lead to very long proofs, potentially even exponential in the size of the original problem. Can we do better? This question guides most of the research in proof complexity.

Most people conjecture that we cannot find short proofs establishing that an arbitrary sudoku puzzle is unsolvable. If we state it in computational complexity terms we get the conjecture that coNP is different from NP. One thing to note is that assuming we could prove this conjecture we would also know that there are no efficient strategies for solving sudoku. For if there existed an efficient strategy

for solving sudoku we could apply it to an unsolvable sudoku. The strategy should then detect the unsolvability of the puzzle and, as we assume that it is an efficient strategy, the description of the steps we took in using this strategy would constitute a short proof of unsolvability of the puzzle. Thus, efficient algorithms for solving sudoku would imply short proofs of unsolvability of sudoku puzzles. Reversing this observation, we have that if there are no short proofs for unsolvability of sudoku then there are no efficient algorithms for solving it, implying that P is not equal to NP. This observation was the original motivation for studying proof complexity.

Currently, we are very far from the goal of proving that there are no short proofs of unsolvability. The reason is that in general there are very little restrictions on how a proof of unsolvability may look like. Moreover, even if we restrict our attention to natural methods of reasoning, we still cannot show that we need long proofs. In order to actually prove some results we need to restrict the reasoning methods even further to the case of very simple systems. In this thesis we focus on a couple of such systems. In the following chapters we will compare two very simple reasoning systems and show how the results in one of the systems can be extended to the other, more powerful system. In the process we will also observe that already in order to make this small step in reasoning power we need to substantially complicate our proof techniques. The next chapter presents formal definitions for the intuitions described here, as well as an overview of the background for this thesis.

Chapter 2

Background

The main concept in computational complexity is that of a Turing machine, an idealized computer that can run any currently known computational process. A Turing machine is a computer with an arbitrary amount of discrete memory, which can be locally manipulated using a finite number of control states. For further details on Turing machines refer to a standard textbook in complexity theory, e.g. [61]. We focus on problems that can be answered by "yes" or "no", that is decision problems. A particular problem can then be identified as a formal language consisting of all instances that have a "yes" answer. A Turing machine then solves a problem if it halts on every input and outputs "true " if the input is in the language and "false" otherwise. There usually exist straightforward transformations between decision problems and problems requiring other kinds of output, such that we can use one solution to efficiently (in polynomial time) solve the other.

We say that a class of problems is efficiently solvable if there is a Turing machine that solves the problem in a polynomial number of steps. This class of problems is known as P. On the other hand, we can also consider problems in which we can verify the solution efficiently. Formally, this is a class of problems such that there is a polynomial time Turing machine that can determine whether a given solution is correct. One requirement is that the solution to the problem is polynomially related to the size of the problem. We can view this solution as a certificate or proof that a solution exists. Note that if there is no solution, then there should not exist any certificate that would be accepted by the verifying Turing machine. This class of problems is called NP.

On the other hand, we can be interested in verifying that there are no solutions to a given problem. In that case, we can ask for a proof/certificate of that claim. This gives us the coNP class of problems. It is easy to see that P is a subset of both NP and coNP. However, we do not know whether all of these classes are distinct or there exists an equality between some of them. These questions are also known as the P vs. NP problem, the most famous problem in theoretical computer science, and the related NP vs. coNP problem. As we intuitively observed in Chapter 1,

proving that NP is distinct from coNP would imply that P is distinct from NP.

Let us now take a closer look at the definition of coNP. We have limited the size of the proof/certificate to be polynomial in the size of the input. We can remove this constraint and require only that the proof verifier runs in the number of steps that is polynomial in the joint size of the input and the proof, while keeping other constraints the same. That is, if the input does not have a solution, then there cannot exist any proof that makes the verifier accept the input. If the input has a solution, then there is at least one proof that is accepted by the verifier. Thus, we require that the proofs are easily verifiable, but do not put any constraints on their size. These constraints correspond to the most general definition of a proof system proposed by Cook and Reckhow [27].

Definition 2.1 (Proof system [27])**.** A *proof system* for a language L is a deterministic algorithm $P(x, \pi)$ that runs in time polynomial in $|x|$ and $|\pi|$ such that

- for all $x \in L$ there is a string π (proof) such that $P(x, \pi)$ outputs "true", and
- for all $x \notin L$ it holds for all strings π that $P(x, \pi)$ outputs "false".

If for a language L and its verifier P there always exists a proof with its size polynomially related to the size of the input, then P is a polynomial proof system. It is straightforward to see that if L is the set of all tautologies of propositional logic, the coNP vs. NP problem turns into the question whether there exists a polynomial proof system for L. The initial goal of proof complexity was to prove that no such proof system exists. However, proving lower bounds for general proof systems is still quite far from what we can currently do. Hence, the current focus of the field is on simpler proof systems for proving tautologies. In these cases it is usually more natural to look at the set of all unsatisfiable formulas instead of tautologies and call proofs refutations. In the rest of the thesis we adopt this view and use the term refutation in order to distinguish the input π for P from our proofs about the behavior of π. We start by looking at one of the simplest proof systems: resolution.

2.1 Resolution

To start we give a brief survey of some of the basic definitions in propositional logic. This is standard material that can be found, e.g., in [55].

A *literal* over a Boolean variable x is either the variable x itself or its negation that is denoted either as $\neg x$ or as $\overline{x}$. We define $\overline{\overline{x}} = x$. A *clause* $C = a_1 \vee \ldots \vee a_k$ is a disjunction of literals and a *term* $T = a_1 \wedge \ldots \wedge a_k$ is a conjunction of literals. We denote the empty clause by $\bot$ and the empty term by $\emptyset$. A clause (term) containing at most k literals is called a *k-clause (k-term)*. A *CNF formula* $F = C_1 \wedge \ldots \wedge C_m$ is a conjunction of clauses. A *DNF formula* $F = T_1 \vee \ldots \vee T_m$ is a disjunction of terms. A *k-CNF formula* is a CNF formula consisting of k-clauses. A *k-DNF formula* is a DNF formula consisting of k-terms. We think of clauses, terms, CNF

and DNF formulas as sets so that order is irrelevant and there are no repetitions. We can now define the resolution proof system introduced by Blake in [18], which Robinson [60] proposed for automated theorem proving. Initial practically efficient search procedures for resolution were proposed by Davis and Putnam [30] and Davis, Logemann, and Loveland [29], and currently resolution is the foundation of most state-of-the-art SAT solvers [4, 49, 53].

Definition 2.2 (Resolution [18])**.** A *resolution configuration* $\mathbb{C}$ is a set of clauses. A *resolution refutation* of a CNF formula F is a sequence of configurations $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$ such that $\mathbb{C}_0 = \emptyset$, $\bot \in \mathbb{C}_\tau$, and for $1 \leq t \leq \tau$ we obtain $\mathbb{C}_t$ from $\mathbb{C}_{t-1}$ by one of the following steps:

Axiom download $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{A\}$, where $A \notin \mathbb{C}_{t-1}$ is a clause in F (sometimes referred to as an *axiom clause*).

Inference $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{D\}$, where $D \notin \mathbb{C}_{t-1}$ is inferred by the *resolution rule* (where G, H denote clauses in $\mathbb{C}_{t-1}$ and x denotes a variable):

$$\frac{G \vee x \qquad H \vee \overline{x}}{G \vee H}$$

Erasure $\mathbb{C}_t = \mathbb{C}_{t-1} \setminus \{D\}$ for $D \in \mathbb{C}_{t-1}$.

The *length* $L(\pi)$ of a resolution refutation π is the number of download and inference steps. The *space* $Sp_{\mathcal{R}}(\pi)$ is the maximal number of clauses in any configuration in π. The *width* $W(\pi)$ is the size of a largest clause in π. We define the length $L(F \vdash \bot)$, the space $Sp_{\mathcal{R}}(F \vdash \bot)$, and the width $W(F \vdash \bot)$ of refuting a formula F in resolution by taking the minimum over all refutations of F with respect to the relevant measure.

An early breakthrough in resolution was the proof of the (sub)exponential lower bound on refutation length for the pigeonhole principle formulas obtained by Haken [44]. Truly exponential lower bounds in the size of the formula were later established in [25, 64]. Essentially all of these bounds were later reproved by Ben-Sasson and Wigderson [14], who identified *width* as a crucial resource. Ben-Sasson and Wigderson proved that strong lower bounds on the width of refutation imply strong lower bounds on the length. This result gives a straightforward way of proving resolution lower bounds, as Ben-Sasson and Wigderson also gave a simple method for proving width lower bounds. However, if the width lower bound is at most a square of the number of variables then this width-length technique does not give any non-trivial lower bounds on length. This is tight as Bonet and Galesi [41] showed that there exist formulas refutable in polynomial length, but requiring quadratic width for their refutation. The relation between length and width notwithstanding, there are formulas for which we can show resolution length lower bounds that cannot use the length-width relation as shown by Dantchev and Riis [28]. The strongest lower bounds to date in terms of the explicit constant in the exponent

were established by Beck and Impagliazzo [8] and further improved by Bonacina and Talebanfard [22].

The study of space in resolution started with Esteban and Torán [34], who gave linear lower bounds for space of Tseitin formulas. It is not too hard to show that this lower bound is tight as space can be at most linear in the formula size. Some further lower bounds on space were proved in [1, 11]. Similarly to the case of length and width, Atserias and Dalmau [3] proved that width is a lower bound for space, again rederiving all then known space lower bounds as corollaries of width lower bounds. However, space is not a lower bound for width as was shown by Ben-Sasson and Nordström [12]. They gave a formula family with constant width complexity but almost linear space complexity. Moreover, Ben-Sasson [10] proved that there exist space-width trade-offs with formulas refutable in constant width and constant space, but such that optimizing one of the measures causes essentially worst-case behaviour of the other. This result was recently strengthened by Berkholz and Nordström [16] who exhibit formulas which can be refuted in both small space and width, but for which any small-width refutation must have space significantly greater than the linear worst-case upper bound.

Instead of only counting the clauses, we can can count all symbols that appear in each clause of a configuration. This measure is called total space. First optimal lower bounds for total space were proved by Bonacina, Galesi and Thapen [21], and later extended by Bennett et al. [15]. Recently, Bonacina [19] showed that width squared is a lower bound for total space, proving a tight relation between width and total space.

We can also ask about connections between length and space. From Atserias and Dalmau [3] it follows that formulas with low space complexity also have short refutations. On the other hand, length is not an upper bound for space as shown in [12]. Nevertheless, if we restrict resolution to the subsystem called *tree-like resolution*, where each line of the refutation can be used only once, Esteban and Torán [34] showed that length upper bounds also imply space upper bounds. Strong trade-offs between length and space in general resolution were proved in [13, 5, 9, 54], showing that there are exist separate refutations in small space and small length, but that both cannot be achieved simultaneously. That is, we can prove exponential lower bounds on refutation length for refutations that have sublinear space [13, 54].

In the next section we explore one proof system that is stronger than resolution: polynomial calculus.

2.2 Polynomial Calculus

In polynomial calculus (or more generally polynomial calculus resolution[1]) we translate a Boolean formula into a set of polynomial equations. As we now deal

[1]In this thesis we use polynomial calculus to refer to both polynomial calculus and polynomial calculus resolution with the distinction being discernable from context. Usually the proof system we refer to is polynomial calculus resolution.

with variables taking values from some field, we need to identify truth values with field elements. Somewhat contrary to intuition, we identify 0 with true and 1 with false. However, this is more natural choice in polynomial calculus. For a field $\mathbb{F}$ we consider the polynomial ring $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$ (where x and $\overline{x}$ are viewed as distinct formal variables). We can now define polynomial calculus resolution as proposed by Alekhnovich et al. [1] extending the original definition of Clegg et al. [26].

Definition 2.3 (Polynomial calculus resolution (PCR) [1, 26])**.** A *PCR configuration* $\mathbb{P}$ is a set of polynomials in $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$. A *PCR refutation* of a CNF formula F is a sequence of configurations $\{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = \emptyset$, $1 \in \mathbb{P}_\tau$, and for $1 \leq t \leq \tau$ we obtain $\mathbb{P}_t$ from $\mathbb{P}_{t-1}$ by one of the following steps:

Axiom download $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is either

- a monomial $m = \prod_{x \in L^+} x \cdot \prod_{y \in L^-} \overline{y}$ encoding a clause $C = \bigvee_{x \in L^+} x \vee \bigvee_{y \in L^-} \overline{y}$ in F, or
- a *Boolean axiom* $x^2 - x$ or *complementarity axiom* $x + \overline{x} - 1$ for any variable x (or $\overline{x}$).

Inference $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is inferred from polynomials $q, r \in \mathbb{P}_t$, variable x, and field elements $\alpha, \beta \in \mathbb{F}$ by either of

- Linear combination
$$\frac{q \qquad r}{\alpha q + \beta r},$$
- Multiplication
$$\frac{q}{xq}.$$

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

If we drop complementarity axioms and encode each negative literal $\overline{x}$ as the polynomial $(1 - x)$, the proof system is called *polynomial calculus (PC)*.

The *size* $S(\pi)$ of a PC/PCR refutation π is the number of monomials (counted with repetitions) in all downloaded or derived polynomials in π, the *(monomial) space* $Sp_{\mathcal{PC}}(\pi)$ is the maximal number of monomials (counted with repetitions) in any configuration in π, and the *degree* $Deg(\pi)$ is the maximal degree of any monomial appearing in π. Taking the minimum over all PCR refutations of a formula F, we define the size $S(F \vdash \bot)$, space $Sp_{\mathcal{PC}}(F \vdash \bot)$, and degree $Deg(F \vdash \bot)$ of refuting F in PCR (and analogously for PC).[2]

If we view polynomial calculus as an extension of resolution then counting monomials instead of polynomials is a natural measure. This holds because each clause of the original formula is transformed into a monomial. Moreover, if we modify

[2]When the proof system is clear from context, we drop the subscript in the notation for space that distinguishes polynomial calculus from resolution.

the definition of polynomial calculus slightly we can show that any k-CNF formula has a refutation in polynomial size if we count only the number of polynomials. For more details on this refer to Paper D. One consequence of the correspondence between clauses and monomials is that the width measure from resolution gets translated to degree in polynomial calculus. With respect to these measures, we have that polynomial calculus simulates resolution with only a small loss in parameters. Moreover, there are formulas for which polynomial calculus can provably do better than resolution.

Compared to resolution, proving lower bounds for size in polynomial calculus is significantly harder. For instance, we do not have any proof techniques for proving size lower bounds without using degree. However, the proof that strong degree lower bounds imply strong size lower bounds was given by Impagliazzo et al. [46]. This proof is analogous to the Ben-Sasson and Wigderson's proof for resolution [14]. Interestingly, the polynomial calculus proof is actually a precursor to the resolution one. Nevertheless, this relation does not resolve the question of size lower bounds as proving degree lower bounds turns out to be much harder than proving resolution width lower bounds. The first polynomial calculus degree lower bound was established by Razborov [57] (later extended in [46] to the size lower bound) for the pigeonhole principle. However, these lower bounds worked with a special encoding of the pigeonhole principle that is not applicable to CNF formulas.

For fields of characteristic distinct from 2, Grigoriev [43] and Buss et al. [24] proposed a technique that performs an affine transformation of the refutation from $\{0, 1\}$ to the "Fourier basis" $\{-1, +1\}$, allowing easier proofs of degree lower bounds. First fully general polynomial calculus degree lower bound that works for any field was proved by Alekhnovich and Razborov [2]. However, their technique was difficult to use and, hence, was followed by only a few further results [40, 41]. Notably, Galesi and Lauria [41] established the optimality of the size-degree relation, mimicking the result of Bonet and Galesi [23] for resolution.

The first space lower bounds in polynomial calculus were proved by Alekhnovich et al. [1], but only sublinear bounds and for formulas of unbounded width. The first space lower bounds for k-CNF formulas were given by Filmus et al. [39], and optimal (linear) lower bounds were proven by Bonacina and Galesi [20]. The latter result was proved for k-CNF formulas where $k \geq 4$, and was later extended to 3-CNF formulas by Bennett et al. [15]. As for the relation between space and degree, it is open whether degree is a lower bound for space (which would be analogue to what holds in resolution). Also, it was previously unknown whether the two measures can be separated in the sense that there are formulas of low degree requiring high space until Paper A presented in this thesis. As for trade-offs between degree and space, Beck et al. [9] proved a space-degree trade-off analogous to the resolution space-width trade-off from [10].

The first trade-off between size and space in polynomial calculus was proved by Huynh and Nordström [45]. However, these were not true trade-offs. They proved that certain formulas have small size refutations and that any refutation in small space must have large size. However, the problem is that we do not know of any

small-space refutations of these formulas and it seems likely that no such refutation exist. The first true trade-off for polynomial calculus was proved by Beck et al. [9] essentially matching the results for resolution except for a small loss in parameters.

We continue by giving a brief overview of the contributions of this thesis. The full papers can be found in Part II.

2.3 Contributions of the Thesis

The first two papers deal with questions related to space and width/degree in resolution and polynomial calculus. The first paper deals with questions about polynomial calculus space and the relation between space and degree. It was coauthored with Yuval Filmus, Massimo Lauria, Jakob Nordström, and Marc Vinyals and was presented at the 40th International Colloquium on Automata, Languages and Programming (ICALP '13) [36]. More details about the paper can be found in Chapter 4 and the full paper is presented as Paper A. The results of the paper are briefly described below:

1. We make progress on the question of whether degree is a lower bound for space in polynomial calculus. We prove that if the resolution width of refuting a CNF formula F is w, then the XORified version $F[\oplus]$ of the formula F requires PCR space $\Omega(w)$. We XORify a formula F by substituting each variable in F with an exclusive or of two new variables and expanding the result out to get a new CNF formula $F[\oplus]$. On one hand, this result is stronger than the claim that degree is a lower bound for space, since small width complexity implies small degree complexity. On the other hand, this is a much weaker result because we need to XORify the formula and we know that XORification can substantially amplify the hardness of a formula. Nevertheless, this is the first and still the only result that makes any connection between width/degree and space in polynomial calculus.

2. Using the previous result, we resolve the other side of the relation between space and degree. We prove essentially optimal separation between degree and space. In order to prove this result we consider (XORified) Tseitin formulas, which encode an unsatisfiable system of linear equations. We show that XORified random Tseitin formulas have proofs of size $\mathrm{O}(n \log n)$ and degree $\mathrm{O}(1)$ in polynomial calculus (even the original one without special variables for negations), but require space $\Theta(n)$ in polynomial calculus resolution. In addition, these small-size proofs are tree-like. Thus, we show that size is not an upper bound on space in tree-like polynomial calculus in contrast to the result in resolution [34].

3. Using ideas related to the ones in previous items allows us to also prove strong PCR space lower bounds for a more general class of Tseitin formulas that have not been XORified. We prove that randomly generated 4-CNF Tseitin

formulas asymptotically almost surely require $\Omega(\sqrt{n})$ space in polynomial calculus to refute.

4. All of the previous results build on the general framework by Bonacina and Galesi [20] for proving polynomial calculus space lower bounds. However, we show that this framework cannot give us all the results that we believe are true in polynomial calculus. Concretely, we show that this framework cannot prove lower bounds for the functional pigeonhole principle formulas, although it seems plausible that these formulas are hard with respect to space.

The second paper, Paper B, in this thesis revisits the space lower bounds in resolution and the relation between space and width. The goal of the paper was to better understand resolution lower bounds in the hope of transporting these insights to polynomial calculus. However, as discussed below, these hopes seem unlikely to be fulfilled. The paper was coauthored with the same set of people as the previous one, Paper A. It was originally published at the 31st Symposium on Theoretical Aspects of Computer Science (STACS '14) [37] and the full version was published in the journal ACM Transactions on Computational Logic [38]. The main results are:

1. We give a new proof of the result by Atserias and Dalmau [3] that width lower bounds space in resolution. They prove that resolution width can be characterized in terms of Ehrenfeucht–Fraïssé games in finite model theory and use this characterization to establish that width is a lower bound for space. On the other hand, our proof of the space-width relation gives a direct combinatorial transformation between small space and small width refutations. That is, we describe a transformation that turns an arbitrary refutation in space s into a refutation that has width at most $s + \mathrm{O}(1)$.

2. With this new proof in hand, we also obtain a new technique for proving space lower bounds in resolution. This new approach is reminiscent of width lower bounds in [14]. We define a static "progress measure" on refutations and argue that when a refutation has made substantial progress (in terms of the defined measure) it must have high space complexity.

3. Finally, we observe that using the new proof of the width-space relation in resolution is unlikely to yield any new insights into polynomial calculus. The problem can be summarized in the observation that a conjunction of variables has a space efficient encoding in polynomial calculus, which is not the case in resolution. This observation leads us to suspect that polynomial calculus has more ways to refute formulas in a space efficient way than resolution has.

The previous two papers shed some light on space and its relation to width/degree in resolution and polynomial calculus. However, the most interesting question of whether degree is a lower bound for space in polynomial calculus still remains open. In the remaining two papers presented in this thesis we take a different track and

move to questions of width/length lower bounds in resolution and degree/size lower bounds in polynomial calculus.

In Paper C, we study formulas that were proposed by Spence and Van Gelder [62, 65] as some of the hardest formulas for current state-of-the-art SAT solvers. The paper was coauthored with Jakob Nordström and published at the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14). A brief description of the results follows:

1. Originally, Spence and Van Gelder [62, 65] introduced what we call subset cardinality formulas and have showed that these formulas are extremely hard experimentally, without any theoretical results corroborating the experiments. In our paper, we rectify that by showing that subset cardinality formulas are exponentially hard in terms of length/size for both resolution and polynomial calculus.

2. We also ran SAT solvers that were state-of-the-art at that time on random instances of subset cardinality formulas, as well as on *fixed bandwidth* formulas that are theoretically easy versions of subset cardinality formulas. We confirmed prior experimental observations. In addition, our experiments also showed that fixed bandwidth formulas are the hardest for SAT solvers, raising the question whether they could be an example of formulas for which current SAT solvers fail to search effectively for resolution refutations.

The aim of the final paper in this thesis, Paper D, was to find a more manageable framework for proving degree and hence size lower bounds in polynomial calculus. The paper is joint work with Jakob Nordström and was published at the 30th Annual Computational Complexity Conference (CCC '15). The main results are as follows:

1. We extend the method of Alekhnovich and Razborov [2] for proving polynomial calculus degree lower bounds. We show that if given a formula F we can construct a graph based on F that satisfies certain properties, then the degree lower bound follows. This extension of the original lower bound method allows us to capture previously known degree lower bounds from [2, 41, 50] in a unified framework. However, there still exist formulas which we believe are hard for polynomial calculus, but where our framework seems inadequate.

2. Using this new framework, we show that functional pigeonhole principle is hard for polynomial calculus, solving one of the open problems Razborov listed in [58].

We now conclude our brief overview of results and move to a more technical discussion. However, before diving into more technical details of the results of this thesis, we make a short digression to discuss the relation between length and width in resolution. The proof that width lower bounds imply length lower bounds will serve as an example of techniques used in proof complexity, as well as let us

(somewhat) complete the picture of the relations that exist between length/size, space, and width/degree in resolution and polynomial calculus.

Chapter 3

Length and Width in Resolution

Ben-Sasson and Wigderson [14], based on work by Impagliazzo et al. [46], proved that strong lower bounds on width of refuting a formula imply strong lower bounds on length. In what follows we denote the width (maximum number of literals in each clause) of the formula F with $W(F)$. For instance, the width of a k-CNF formula is equal to k. We now formally state the length-width relation in resolution.

Theorem 3.1 (Ben-Sasson and Wigderson [14])**.** *The length of refuting a CNF formula F over n variables in resolution is bounded from below by*

$$L(F \vdash \bot) = \exp\left(\Omega\left(\frac{(W(F \vdash \bot) - W(F))^2}{n}\right)\right).$$

The main idea of the proof is to take a refutation in small length, break it apart into different pieces and then stitch the pieces back together to produce a refutation in small width. In the process the length of the refutation will blow-up substantially, but we will get the desired small width. In order to facilitate achieving this goal we rewrite Theorem 3.1 as an upper bound on the width of refuting a formula F, reintroducing constants not present in Theorem 3.1.

Lemma 3.2. *The width of refuting a CNF formula F over n variables in resolution is bounded from above by*

$$W(F \vdash \bot) \leq \max\left\{W(F), \left\lceil\sqrt{2n \ln L(F \vdash \bot)}\right\rceil\right\} + \left\lceil\sqrt{2n \ln L(F \vdash \bot)}\right\rceil. \quad (3.1)$$

It is not hard to see that by replacing the maximum operator with summation and rearranging the inequality we get back Theorem 3.1.

As mentioned previously, we will break apart the short length refutation and stitch it back together to produce a new one. In breaking apart the proof we use *restrictions*, that is *partial assignments* ρ to the variables of the formula F. In a restricted formula $F\restriction_\rho$ (or refutation $\pi\restriction_\rho$) all clauses satisfied by ρ are removed

and all other clauses have falsified literals removed. Restrictions preserve both resolution and polynomial calculus refutations (up to some minor modifications of the refutation). Hence, if π is a refutation of F, then $\pi\restriction_\rho$ is a refutation of $F\restriction_\rho$ in at most the same length/size, width/degree, and space (except possibly for a constant factor in size in polynomial calculus due to postprocessing steps). For two restrictions ρ and ρ' over distinct domains, we denote by $\rho \cup \rho'$ the restriction ρ'' such that $\rho''(x) = \rho(x)$ or $\rho''(x) = \rho'(x)$ depending on whether x belongs to the domain of ρ or ρ', respectively.

We will break apart our small length refutation by producing refutations of two different formulas $F\restriction_x$ and $F\restriction_{\overline{x}}$, where we use x to denote restriction ρ that sets $\rho(x) = \top$ and similarly use $\overline{x}$ to denote $\rho(x) = \bot$. To simplify the following proofs we add one more inference rule to resolution: *weakening*. In weakening we can infer the clause $C \vee D$ from C for an arbitrary clause D. It is not hard to see that weakening steps can be removed from any refutation without any loss in complexity. To begin our proof, we show how to stitch back together the two different refutations of $F\restriction_x$ and $F\restriction_{\overline{x}}$.

Lemma 3.3. *For a literal l, if $W(F\restriction_l \vdash \bot) \leq w - 1$ and $W(F\restriction_{\bar{l}} \vdash \bot) \leq w$ then it holds that $W(F \vdash \bot) \leq \max\{w, W(F)\}$.*

Proof. First, we show that from the refutation $\pi : F\restriction_l \vdash \bot$ with $W(F\restriction_l \vdash \bot) \leq w-1$ we can construct a derivation $\pi' : F \vdash \bar{l}$ that has width at most w. The main idea is to follow the refutation π using the axioms from F instead of $F\restriction_l$. This means that a particular axiom A in $F\restriction_l$ might turn into $A \vee \bar{l}$ in F (as those are the ones that get truncated by restricting F). Following this process further, it is not hard to see it results with π' where each clause is of the form C or $C \vee \bar{l}$ for the corresponding clause C in π. Thus, we get that the final clause of π' is either the empty clause $\bot$ or $\bar{l}$, where in the former case we can just use weakening to derive $\bar{l}$. As we have added at most one literal to each clause of π, we have shown that $W(F \vdash \bar{l}) \leq w$.

With $\bar{l}$ in hand from π', we resolve out literal l from all axioms in F that contain it. This results in essentially the formula $F\restriction_{\bar{l}}$. Let us denote this part of refutation by π''. Note that π'' has width upper bounded by the width of the formula $W(F)$. We can now just run the refutation $\pi''' : F\restriction_{\bar{l}} \vdash \bot$ in width w, which exists by assumption, to construct the final refutation of F. That is, piecing together derivations π', π'', and π''' in sequence produces a refutation of F with width at most $W(F \vdash \bot) \leq \max\{w, W(F)\}$. □

The main part of the proof of Lemma 3.2 consists of reducing the width of clauses that appear in the small length refutation. The clauses that we focus on are the "fat" clauses that have width lower bounded by some threshold $d, d \geq 1$. For an arbitrary refutation π we denote by $\mathrm{fat}_d(\pi)$ the number of clauses C in π such that $W(C) \geq d$. The next lemma shows us how we can trade-off the number of fat clauses in a refutation for a refutation of smaller width. We simplify the notation and use $\rho \cup l$ to denote the restriction $\rho \cup \{l\}$, where we assume that ρ does not set a variable corresponding to the literal l.

Lemma 3.4. *Let $d \geq 1$ be an integer and π a refutation of a CNF formula F. If there exists a real-valued constant $a > 1$ such that for any restriction ρ there exists a literal l in the variables of $F\restriction_\rho$ with $\mathrm{fat}_d(\pi\restriction_{\rho\cup l}) \leq \frac{1}{a}\,\mathrm{fat}_d(\pi\restriction_\rho)$, then*

$$W(F \vdash \bot) \leq \max\{\, W(F), d\} + \lceil \log_a \mathrm{fat}_d\,(\pi) \rceil\,,$$

where we assume that $\log_a \mathrm{fat}_d(\pi) = 0$ when $\mathrm{fat}_d(\pi) = 0$.

Proof. Let ρ be an arbitrary restriction including an empty one. We use a nested induction over the number of fat clauses $\mathrm{fat}_d(\pi\restriction_\rho)$ and the number of variables of $F\restriction_\rho$ to show that $W(F\restriction_\rho \vdash \bot) \leq \max\{\, W(F\restriction_\rho), d\} + \lceil \log_a \mathrm{fat}_d(\pi\restriction_\rho) \rceil$. First, for the basis of the induction we have that if $\mathrm{fat}_d(\pi\restriction_\rho) = 0$ then $\pi\restriction_\rho$ is a refutation of $F\restriction_\rho$ that has all clauses with width strictly less than d. Thus the inductive bound is satisfied. Otherwise, if the number of variables of $F\restriction_\rho$ is equal to 0 we have that the formula $F\restriction_\rho$ consists only of the empty clause and there are no fat clauses in its refutation and hence the first case holds.

Now we show that the lemma still holds for a restriction ρ such that the number of variables of $F\restriction_\rho$ and the number of fat clauses $\mathrm{fat}_d(\pi\restriction_\rho)$ are both strictly greater than 0. By assumption there exists a literal l over the variables of $F\restriction_\rho$ such that $\mathrm{fat}_d(\pi\restriction_{\rho\cup l}) \leq \frac{1}{a}\,\mathrm{fat}_d(\pi\restriction_\rho)$. By induction we have

$$\begin{aligned} W(F\restriction_{\rho\cup l} \vdash \bot) &\leq \max\{\, W(F\restriction_{\rho\cup l}), d\} + \lceil \log_a \mathrm{fat}_d\,(\pi\restriction_{\rho\cup l}) \rceil && (3.2)\\ &\leq \max\{\, W(F\restriction_\rho), d\} + \left\lceil \log_a \frac{1}{a}\,\mathrm{fat}_d\,(\pi\restriction_\rho) \right\rceil && (3.3)\\ &\leq \max\{\, W(F\restriction_\rho), d\} + \lceil \log_a \mathrm{fat}_d\,(\pi\restriction_\rho) \rceil - 1. && (3.4) \end{aligned}$$

On the other hand, for $F\restriction_{\rho\cup\bar{l}}$ we know only that the number of variables in $F\restriction_\rho$ got reduced, while the number of fat clauses in $\pi\restriction_\rho$ might have stayed the same. Hence, we can apply induction to get:

$$\begin{aligned} W(F\restriction_{\rho\cup\bar{l}} \vdash \bot) &\leq \max\left\{\, W(F\restriction_{\rho\cup\bar{l}}), d\right\} + \left\lceil \log_a \mathrm{fat}_d\left(\pi\restriction_{\rho\cup\bar{l}}\right) \right\rceil && (3.5)\\ &\leq \max\{\, W(F\restriction_\rho), d\} + \lceil \log_a \mathrm{fat}_d\,(\pi\restriction_\rho) \rceil\,. && (3.6) \end{aligned}$$

Now, we apply Lemma 3.3 to conclude that

$$W(F\restriction_\rho \vdash \bot) \leq \max\{\, W(F\restriction_\rho), d\} + \lceil \log_a \mathrm{fat}_d\,(\pi\restriction_\rho) \rceil\,. \qquad (3.7)$$

We get the final result of the lemma by taking ρ to be the empty restriction, thereby operating on the vanilla formula F and refutation π. □

The previous lemma allows us to exchange the refutation with a small number of fat clauses for a refutation having small width. The crucial part is identifying a literal l which significantly reduces the number of fat clauses. Hence, we need to identify the best value of the reduction factor a that we can achieve for an arbitrary formula. The following lemma gives us one good bound.

Lemma 3.5. *For an integer $d \geq 1$ and a refutation $\pi : F \vdash \bot$ of a formula F over n variables, we have that there exists a literal l such that*

$$\mathrm{fat}_d(\pi\restriction_l) \leq \left(1 - \frac{d}{2n}\right) \mathrm{fat}_d(\pi)\,.$$

Proof. First note that if $d > 2n$ then $\mathrm{fat}_d(\pi) = 0$. This holds because there are $2n$ different literals over n variables implying that any clause in π has width at most $2n$. In this case the bound in the lemma trivially holds as both sides of the inequality are equal to 0.

Otherwise, as each fat clause in π has at least d literals, there are at least $d \cdot \mathrm{fat}_d(\pi)$ literals in the fat clauses of π (possibly with repetitions). We estimate the minimal number of times that the most frequently occurring literal appears in the fat clauses of π. As there are $2n$ different literals, we have that there is a literal l that appears in at least $\frac{d}{2n}\,\mathrm{fat}_d(\pi)$ fat clauses of π. Setting that literal to true satisfies all such clauses that contain l. Hence, those clauses do not exist in $\pi\restriction_l$ and the number of fat clauses is bounded by

$$\mathrm{fat}_d(\pi\restriction_l) \leq \left(1 - \frac{d}{2n}\right) \mathrm{fat}_d(\pi)\,, \tag{3.8}$$

proving the lemma. □

Now, we can put the pieces together to produce the proof of Lemma 3.2.

Proof of Lemma 3.2. We need to show that for any CNF formula F over n variables it holds that $W(F \vdash \bot) \leq \max\{W(F), \sqrt{2nL(F \vdash \bot)}\} + \sqrt{2nL(F \vdash \bot)}$. Let $\pi : F \vdash \bot$ be a refutation of F that achieves the optimal length bound $L(F \vdash \bot)$ and denote $L = L(\pi) = L(F \vdash \bot)$. It holds that $\mathrm{fat}_d(\pi) \leq L$. Now, we use Lemma 3.5 to get the constant a, which we will then plug into Lemma 3.4.

We set $a = \left(1 - \frac{d}{2n}\right)^{-1}$, where we require that $1 \leq d < 2n$. It follows that $a > 1$ satisfying one condition of Lemma 3.4. By Lemma 3.5 we have that for any restriction ρ there is a literal l in $F\restriction_\rho$ such that $\mathrm{fat}_d(\pi\restriction_{\rho\cup l}) \leq \left(1 - \frac{d}{2m}\right) \mathrm{fat}_d(\pi\restriction_\rho)$ where m is the number of variables in $F\restriction_\rho$. As $m \leq n$ for such an l it also holds that $\mathrm{fat}_d(\pi\restriction_{\rho\cup l}) \leq \left(1 - \frac{d}{2n}\right) \mathrm{fat}_d(\pi\restriction_\rho)$. Hence, our choice of a also satisfies the second condition of *Lemma* 3.4 and we can deduce that

$$W(F \vdash \bot) \leq \max\{W(F), d\} + \lceil \log_a L \rceil\,, \tag{3.9}$$

where $a = \left(1 - \frac{d}{2n}\right)^{-1}$, as $\mathrm{fat}_d(\pi) \leq L$.

To find the best value for d we write $\log_a L$ as $\ln L / \ln a$ and lower bound $\ln a$. We have

$$\ln a = \ln\left(1 - \frac{d}{2n}\right)^{-1} = -\ln\left(1 - \frac{d}{2n}\right) \geq \frac{d}{2n}, \tag{3.10}$$

as $\ln(1 + x) \leq x$ whenever $x \geq -1$. Thus $\log_a L \leq \frac{2n \ln L}{d}$. To set d we minimize the expression $d + \frac{2n \ln L}{d}$, as this gives us the tightest upper bound on width up

to additive constants. The optimal setting is $d = \lceil \sqrt{2n \ln L} \rceil$. As length is strictly less than 2^{n+1} (which can be showed for any formula F over n variables) we have that $d \leq \sqrt{2n(n+1)\ln 2} < 2n$ when $n \geq 1$, satisfying the condition on d. Hence our final bound on the width of refuting F is

$$W(F \vdash \bot) \leq \max \left\{ W(F), \left\lceil \sqrt{2n \ln L(F \vdash \bot)} \right\rceil \right\} + \left\lceil \sqrt{2n \ln L(F \vdash \bot)} \right\rceil , \quad (3.11)$$

proving Lemma 3.2. □

A similar proof also works for polynomial calculus and the relation between size and degree. Thus, this proof gives us a relation between length/size and width/degree in resolution and polynomial calculus. In the rest of the thesis we survey papers that elaborate more on the relations between space and width/degree, as well as prove width/degree lower bounds. The first paper in the sequence deals with space in polynomial calculus.

Chapter 4

Paper A. Towards an Understanding of Polynomial Calculus

In the paper "Towards an Understanding of Polynomial Calculus: New Separations and Lower Bounds" [36], we explore the questions about space in polynomial calculus. The paper builds on the work by Bonacina and Galesi [20] that presents a framework for proving polynomial calculus lower bounds. In this chapter, we present two main results of the paper that relate space and degree in polynomial calculus, sketching the proof of one of them. At the end of the chapter we survey the remaining results of the paper.

4.1 Space and Degree in Polynomial Calculus

The central result we establish presents partial progress on understanding the relation between space and degree in polynomial calculus. We show that if the resolution width of refuting a CNF formula F is large, then by XORifying F we obtain the formula $F[\oplus]$ that requires large polynomial calculus space. The notation $F[\oplus]$ denotes the CNF formula we obtain by substituting every variable x in F with $x_1 \oplus x_2$, where x_1 and x_2 are new variables for every x, and expanding out such a formula to conjunctive normal form. Formally we prove the following theorem.

Theorem 4.1. *For a k-CNF formula F it holds over any field that*

$$Sp_{\mathcal{PCR}}(F[\oplus] \vdash \bot) = \Omega(W_{\mathcal{R}}(F \vdash \bot)).$$

The main idea of the theorem is to combine the framework for space lower bounds in polynomial calculus by Bonacina and Galesi [20] with the characterization of resolution width by Atserias and Dalmau [3]. An almost immediate consequence of this theorem is that there are formulas that have polynomial calculus refutations

in constant degree but nevertheless require maximal space. Here, we sketch out the proof for the field of characteristic 2. First, let us define the formulas that are crucial for this result.

Definition 4.2 (Tseitin formula)**.** Let $G = (V, E)$ be an undirected graph and $\chi : V \to \{0, 1\}$ be a function. Identify every edge $e \in E$ with a variable x_e and let $PARITY_{v,\chi}$ denote the CNF encoding of the constraint that the number of true edges x_e incident to a vertex $v \in V$ is equal to $\chi(v) \pmod 2$. Then the *Tseitin formula* over G with respect to f is $T_{G,\chi} = \bigwedge_{v \in V} PARITY_{v,\chi}$.

We use these formulas to prove our separation between space and degree in polynomial calculus.

Theorem 4.3. *For polynomial calculus over $\mathbb{F}_2$, there is a family of k-CNF formulas F_n of size $\mathrm{O}(n)$ such that $Sp_{\mathcal{PCR}}(F_n \vdash \bot) = \Omega(n)$ but which have polynomial calculus refutations with $Deg(\pi_n) = \mathrm{O}(1)$.*

Proof sketch. For each n, we set F_n to be a Tseitin formula $T_{G,\chi}$ over an expander graph (a graph with very good connectivity) with n vertices. From Ben-Sasson and Wigderson [14] we know that $W_{\mathcal{R}}(F_n \vdash \bot) = \Omega(n)$ for such graphs.

If we now take $F_n[\oplus]$ instead, by Theorem 4.1 we get the polynomial calculus space lower bound. On the other hand, it is not hard to see that XORification yields another Tseitin formula as we can interpret XOR as turning the graph into a multi-graph by doubling each edge. The degree upper bound for $F[\oplus]$ then follows by observing that unsatisfiable systems of linear equations can easily be refuted by summing up all equations. □

4.2 Other Results

Looking more carefully at the proof of Theorem 4.3, we extend the result on Tseitin formulas from multi-graphs to d-regular graphs with $d \geq 4$. Formally, we prove the following theorem.

Theorem 4.4. *Let G be a random d-regular graph on n vertices, where $d \geq 4$. Then over any field it holds almost surely that $Sp_{\mathcal{PCR}}(T_{G,\chi} \vdash \bot) = \Omega\big(\sqrt{n}\big)$.*

As all of the previous theorems used Bonacina and Galesi's framework [20] for proving space lower bounds, we can ask whether this framework allows us to prove all space lower bounds that we care about. Unfortunately this does not seem to be the case if we consider the functional pigeonhole principle, a particular encoding of a statement that $n + 1$ pigeons cannot nest in n holes (defined later in Chapter D). While these formulas need large degree to refute [57, 51] and we believe that they require large space as well, the current framework cannot establish that result as shown by our final theorem of this paper.

Theorem 4.5. *There is no r-extendible family for $FPHP_n^{n+1}$ for $r > 1$.*

Chapter 5

Paper B. From Small Space to Small Width in Resolution

In the paper "From Small Space to Small Width in Resolution" [38], we study the relation between space and width in resolution. As seen before, Atserias and Dalmau [3] showed that width is a lower bound on space in resolution. The basis of their paper was a combinatorial characterization of the resolution width as a particular kind of Ehrenfeucht–Fraïssé game. With this alternative characterization in hand, they showed that any small space refutation in resolution can be viewed as an efficient strategy for this game. While their paper gave us a better understanding of resolution width, the proof of the space-width relation felt a bit opaque. In this paper we improve on this state of affairs by giving a direct translation from small space into small width refutations in resolution. Moreover, we also present a new proof technique for proving space lower bounds in resolution building on this new proof of the space-width relation.

5.1 The New Proof of Space-Width Relation in Resolution

Here we sketch our new proof that space upper bounds width in resolution.[1]

Theorem 5.1 ([3]). *Let $\pi : F \vdash \bot$ be a resolution refutation of a k-CNF formula F in space $Sp(\pi) = s$. Then there exists a resolution refutation π' of F in width $W(\pi') \leq s + k - 3$.*

In our proof we start with a resolution refutation in small space written out as a sequence of configurations and then negate each configuration in the refutation. As a contradiction turns into a tautology and vice versa under negation we have to run the new refutation backwards. The rest of the proof consists in filling in the details in order to make sure that the new refutation is a legal one and has the right width upper bound.

[1] Razborov also found a similar proof, but did not publish it.[56]

Definition 5.2. The *negated configuration* $\text{neg}(\mathbb{C})$ of a configuration $\mathbb{C}$ is defined by induction on the number of clauses in $\mathbb{C}$:

- $\text{neg}(\emptyset) = \{\bot\}$,
- $\text{neg}(\mathbb{C} \cup \{C\}) = \{D \vee \overline{a} \mid D \in \text{neg}(\mathbb{C}) \text{ and } a \in C\}$,

where we remove trivial and subsumed clauses from the final configuration.

We want to take a resolution refutation $\pi = (\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau)$ and prove that if it has small space, then the reversed sequence of negated configurations $\pi' = (\text{neg}(\mathbb{C}_\tau), \text{neg}(\mathbb{C}_{\tau-1}), \ldots, \text{neg}(\mathbb{C}_0))$ has small width. As π' is not necessarily a legal refutation, we need to show how to derive the clauses in each configuration of the negated refutation. Inference and clause deletion steps turn out to be easy to run in reverse. They follow from the fact that in inference and clause deletion we have that $\mathbb{C}_i \vDash \mathbb{C}_{i+1}$. In such cases it is not hard to see that for every clause $C \in \text{neg}(\mathbb{C}_i)$ there is a clause $C' \in \text{neg}(\mathbb{C}_{i+1})$ such that C is a weakening of C'.[2] Thus, the clauses in the negated configuration can be derived easily.

The axiom download case is a bit harder to prove and it introduces the k from k-CNF in the upper bound in Theorem 5.1. The basic analysis here consists of noting that for a clause $C \in \text{neg}(\mathbb{C}_i)$ the configuration $\text{neg}(\mathbb{C}_{i+1})$ essentially contains clauses $C_a = C \vee \overline{a}$ for all literals a in the downloaded axiom A. Thus, we can use resolution over A and clauses C_a to derive the clause C in the reverse refutation. The details can be found in the full paper.

The hope in finding this new proof was that it would help us prove the space-degree relation in polynomial calculus. However, this seems unlikely. An example of formulas that seem hard to deal with in this way are *pebbling contradictions*. Pebbling contradictions are defined in terms of directed acyclic graphs (DAGs) with a unique sink (vertex with no outgoing edges). The formulas state that each source (vertex with no incoming edges) is true, that each inner vertex is true if its parents are true, and that the sink is false.

The main observation is that in resolution there are two natural refutations of pebbling contradictions, which are reverse images of each other. One keeps a big AND of literals (a small width refutation), while the other keeps a big OR (a small space refutation). Negating one of these refutations produces the other. The complication in polynomial calculus is that AND can be in small space. Thus we have a second small space refutation that does not exist in resolution, while it does not seem likely that we can produce a second small degree refutation. Therefore, the proof of space-degree relation in polynomial calculus would have to introduce some fundamental changes to our proof technique in order for us to have any hope of making it work.

[2] See Chapter 3 for the discussion of the weakening rule.

Chapter 6

Paper C. Long Proofs of (Seemingly) Simple Formulas

In the paper "Long Proofs of (Seemingly) Simple Formulas" [50] we leave the questions of space and its relation to width/degree and return to the questions of length/size lower bounds. We explore the connection between theoretical results for resolution and experimental results for *conflict-driven clause learning (CDCL)* SAT solvers. The basis for this exploration is the class of formulas proposed by Van Gelder and Spence [62, 65], which we call *subset cardinality formulas*. Van Gelder and Spence showed that subset cardinality formulas were among the hardest formulas for then current state-of-the-art SAT solvers, but they did not provide a theoretical justification for that fact. We make progress on this question by proving that subset cardinality formulas are hard for both resolution and polynomial calculus. In addition, we further explore the experimental hardness of these formulas and add some new observations about the state-of-the-art SAT solvers.

6.1 Theoretical Hardness of Subset Cardinality Formulas

To form subset cardinality formulas we start with a set of $4n + 1$ variables, which are (randomly) partitioned into groups of 4 plus one group of 5 variables. For each of these groups we write down clauses encoding the constraint that at least half of the variables in the group are true, that is 2 variables for 4-groups and 3 variables for a 5-group. Furthermore, we take a second random partition into groups of 4 and one group of 5, but now encode the constraint that at least half of the variables are false. By a counting argument it is not hard to see that such formulas must be unsatisfiable. There are a few ways that we can improve on this construction in order to ensure that an average formula is even harder for SAT solvers. For instance, we can base the formula on 4-regular bipartite graphs with an extra edge added. We define this kind of formula next.

Definition 6.1 (Subset cardinality formula)**.** Suppose that $G = (U \dot\cup V, E)$ is a 4-regular bipartite graph except that one extra edge has been added. Then the *subset cardinality formula* $SC(G)$ over G has variables $x_e, e \in E$, and clauses:

- $x_{e_1} \vee x_{e_2} \vee x_{e_3}$ for every triple e_1, e_2, e_3 of edges incident to $u \in U$,
- $\overline{x}_{e_1} \vee \overline{x}_{e_2} \vee \overline{x}_{e_3}$ for every triple e_1, e_2, e_3 of edges incident to $v \in V$.

The main theoretical result in the paper is that if the bipartite graph is chosen at random, we are almost sure to produce a hard formula for resolution and polynomial calculus. To prove this result we use a restriction that when applied to a subset cardinality formula produces a pigeonhole principle formula.[1] As these formulas have exponential lower bounds in resolution and polynomial calculus, our theorem follows. The restriction that is used is based on an arbitrary matching, which must exist in such a graph, by setting the edges in the matching to true. The 5-degree vertices are dealt with separately. We have the following theorem.

Theorem 6.2. *The formula $SC(G)$ for a random 4-regular bipartite graph G with an arbitrary extra edge added requires polynomial calculus refutations (and hence also resolution refutations) of exponential size asymptotically almost surely.*

6.2 Experimental Results

We also ran experiments with SAT solvers Glucose 2.2 [42], March-rw [48], and Lingeling-ala [47] on subset cardinality formulas, as well as a few benchmark formulas: random 3-CNFs and Tseitin formulas. An example of the results can be seen in Figure 6.1. As this and the other experiments presented in the full paper indicate, the subset cardinality formulas are among the hardest formulas for modern SAT solvers. However, we also see an interesting phenomenon where the theoretically easiest formulas, the fixed bandwidth formulas, are the hardest for SAT solvers (they time-out on instances with the smallest number of variables). These formulas are a special kind of subset cardinality formulas where the neighborhood relations of the graph follow a specific pattern, which makes formulas easy to refute.

To further explore this issue, we also ran experiments where we fixed the order in which the SAT solver branches on different variables. For this we have used a modified MiniSat 2.2.0 [52]. An example of the results can be seen in Figure 6.2. With a better ordering the results for fixed-bandwidth formulas improve, bringing them in line with the theoretical understanding. Moreover, using a different ordering, for instance one suggested by Elffers [31], gives significantly better results than the ones produced by our ordering. This further confirms that fixed bandwidth formulas can be made easy for SAT solvers. An interesting, although speculative, question that these experiments raise is whether fixed-bandwidth formulas could be used to

[1] Formally, we reduce it to a special kind of pigeonhole principle formulas that are based on well connected graphs.

formally show that CDCL with current heuristics does not polynomially simulate resolution. Note also that the advances in pseudo-Boolean solvers reported in [17] show that subset cardinality formulas can be easy when the reasoning system can detect and use cardinality constraints.

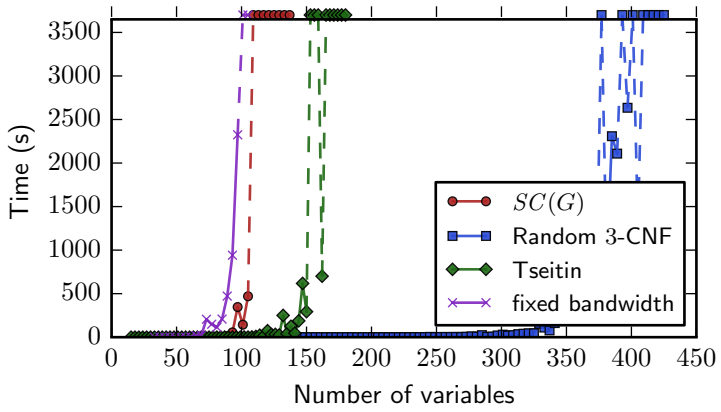


Figure 6.1: Comparison of subset cardinality formulas with other benchmarks.

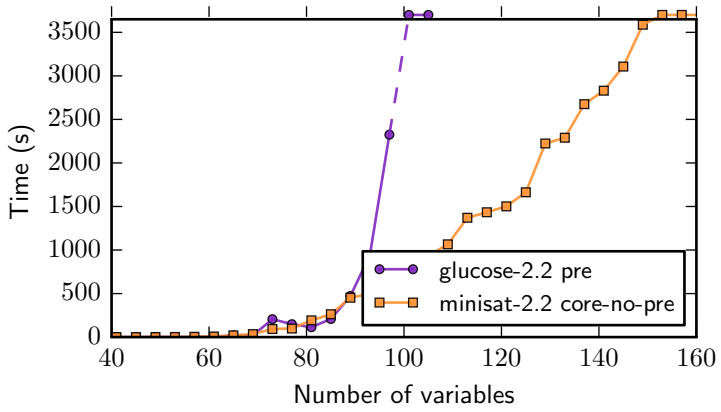


Figure 6.2: Solving fixed-bandwidth formulas using fixed variable ordering.

Chapter 7

Paper D. A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds

Paper "A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds" [51] explores the question of degree lower bounds in polynomial calculus. By a version of Theorem 3.1 for polynomial calculus we have that we can translate degree lower bounds into size lower bounds. However, unlike resolution, in polynomial calculus we do not have a well-developed machinery for proving degree lower bounds. In this paper we improve on this situation by introducing a graph structure that, if it can be built from the CNF formula, implies the degree lower bound. An even more general framework was independently developed by Filmus [35]. Filmus gives different, more explicit, proofs of the key technical lemmas in [2], but does not obtain any new lower bound results.

In this chapter we give an overview of a simplified version of our graph framework for proving degree lower bounds. We also present a brief discussion on how this framework differs from the framework that allows us to prove resolution width lower bounds. In the second part of the chapter we give an overview of different versions of pigeonhole principle formulas and our contributions to resolving their hardness.

7.1 A Generalized Clause-Variable Incidence Graph

We build a bipartite graph representing the CNF formula $\mathcal{F}$ by splitting the formula into subformulas (i.e., subsets of clauses). That is, we take a family $\mathcal{U}$ of subformulas F of $\mathcal{F}$ turning each subformula F into a vertex on the left-hand side of the graph. We also partition the variables of $\mathcal{F}$ into a family $\mathcal{V}$ of subsets of variables V to get the vertices on the right-hand side of the graph. We place an edge between a formula F and a set of variables V if they share at least one variable. An important requirement for this graph is that for each edge (F, V) we can satisfy the formula F

by setting only the variables in V. In what follows, we use the notation $\mathit{Vars}(F)$ to denote the set of all variables that appear in F.

Definition 7.1 (Bipartite $(\mathcal{U},\mathcal{V})$-graph)**.** Let $\mathcal{U}$ be a set of CNF formulas, and $\mathcal{V}$ be a partition of the set of variables $\bigcup_{F\in\mathcal{U}} \mathit{Vars}(F)$. Then the $(\mathcal{U},\mathcal{V})$-*graph* is a bipartite graph with left vertices $F \in \mathcal{U}$, right vertices $V \in \mathcal{V}$, and edges between F and V if $\mathit{Vars}(F) \cap V \neq \emptyset$. Furthermore, for every edge (F, V) we require that there is an assignment ρ to variables in V that satisfies F.

We use standard graph notation and write $N(F)$ to denote the set of all neighbours $V \in \mathcal{V}$ of a vertex/CNF formula $F \in \mathcal{U}$. The crucial criteria for determining the hardness of the formula $\mathcal{F}$ is whether a $(\mathcal{U},\mathcal{V})$-graph that we build from $\mathcal{F}$ is expanding, meaning that the sets of vertices on the left-side of the graph have a lot of unique neighbors on the right. The formal definitions follow.

Definition 7.2 (Boundary of a $(\mathcal{U},\mathcal{V})$-graph)**.** For a $(\mathcal{U},\mathcal{V})$-graph and a subset $\mathcal{U}' \subseteq \mathcal{U}$, the *boundary* $\partial(\mathcal{U}')$ of $\mathcal{U}'$ is the family of variable sets $V \in \mathcal{V}$ such that each $V \in \partial(\mathcal{U}')$ is a neighbor of some clause set $F \in \mathcal{U}'$ but is not a neighbor of any other clause set $F' \in \mathcal{U}' \setminus \{F\}$.

Definition 7.3 (Boundary expander)**.** A $(\mathcal{U},\mathcal{V})$-graph is said to be an (s,δ)-*boundary expander* if for every set $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$, it holds that $|\partial(\mathcal{U}')| \geq \delta|\mathcal{U}'|$.

With the definitions above we can state the simplified version of our main theorem on degree lower bounds in polynomial calculus.

Theorem 7.4. *Let a $(\mathcal{U},\mathcal{V})$-graph be an (s,δ)-boundary expander. Then any polynomial calculus refutation of $\bigwedge_{F\in\mathcal{U}} F$ requires degree strictly greater than $\delta s/2$.*

Recall that we required that for each edge (F, V) there exists an assignment ρ to V that satisfies F. Another way of viewing this statement is through a kind of "edge game" played against an adversary. In this game we start first by setting the variables in V, after which the adversary sets the remaining variables to whichever values he wants. We win the game if F is satisfied after both we and the adversary make our moves. Then the constraint that V must satisfy F translates into the requirement that we can always win the "edge game".

The difference in resolution is that we change the "edge game" so that the adversary is required to go first and set all variables outside of V. When adversary finishes his step we can proceed to set the variables in V however we like in order to achieve the same goal of satisfying F. This gives us more power in resolution and makes the game easier, as we do not need to anticipate all possible moves made by the adversary. One example where this makes a difference are the Tseitin formulas from Definition 4.2. Taking each formula $F_v \in \mathcal{U}$ to be an encoding of a constraint $\mathit{PARITY}_{v,\chi}$ on a vertex v and $\mathcal{V}$ consisting of singleton sets of one variable each we can form the $(\mathcal{U},\mathcal{V})$-graph for Tseitin. It is not hard to see that the "edge game"

for resolution is winnable, while the polynomial calculus game is not. This is in line with the fact that in polynomial calculus we can efficiently refute Tseitin formulas.

The framework presented in this chapter is a significantly simplified version of the original framework and it does not allow us to prove lower bounds for many formulas. The main concern is that an original formula might not consist only of the subformulas that can be arranged into an expanding graph. If that is the case we isolate the non-expanding part of the formula into a separate subformula E that is not an immediate part of the graph, but which changes the "edge game". If we have the subformula E, all of the assignments in the "edge game" must not falsify E. That is, the adversary's assignment must not falsify E, while the union of our and the adversary's assignment must satisfy both the formula F that belongs to the edge as well as E. In the paper, we also distinguish between the edges on which we can win this new "edge game" and the edges on which we cannot, and define the expansion accordingly. However, we do not need this distinction in any of our applications.

The second distinction between the simplified framework and the full framework is that we do not require $\mathcal{V}$ to be a partition, but allow some variables to appear in multiple sets. In this case we need to bound the number of sets in which a variable can appear and this then weakens the lower bound in Theorem 7.4. Nevertheless, this modification is needed to show lower bounds for ordering principle formulas, originally shown hard for degree in [41], and functional pigeonhole principle formulas. In the next section we survey the bound on pigeonhole principle formulas and how our results fit into them.

7.2 Pigeonhole Principle Bounds

We start by giving a formal definition of different versions of pigeonhole principle formulas, using the notation $[n] = \{1, 2, \ldots, n\}$. The *pigeonhole principle formulas* are CNF formulas over variables $x_{p,h}$, $p \in [n+1]$ and $h \in [n]$, which we interpret as being true if pigeon p nests in hole h. We have the following axioms:

$$\bigvee_{h=1}^{n} x_{p,h} \qquad p \in [n+1] \qquad \text{(pigeon axioms)} \tag{7.1a}$$

$$\overline{x}_{p,h} \vee \overline{x}_{p',h} \qquad h \in [n],\ p, p' \in [n+1],\ p \neq p', \qquad \text{(hole axioms)} \tag{7.1b}$$

$$\overline{x}_{p,h} \vee \overline{x}_{p,h'} \qquad p \in [n+1],\ h, h' \in [n],\ h \neq h' \qquad \text{(functionality axioms)} \tag{7.1c}$$

$$\bigvee_{p=1}^{n+1} x_{p,h} \qquad h \in [n] \qquad \text{(onto axioms)} \tag{7.1d}$$

The standard *pigeonhole principle formula* PHP_n^{n+1} is the formula consisting of only the pigeon and hole axioms. *The functional pigeonhole principle formula* $FPHP_n^{n+1}$ is the pigeonhole principle formula PHP_n^{n+1} with functional axioms added, the *onto pigeonhole principle formula Onto-*PHP_n^{n+1} is PHP_n^{n+1} with onto axioms added,

Table 7.1: Comparison of pigeonhole principle formulas in resolution and polynomial calculus. Hard denotes an exponential lower bound, while easy denotes a polynomial upper bound in the number of holes n.

Variant	Resolution	Polynomial Calculus
PHP_n^{n+1}	hard [44]	hard [2]
$FPHP_n^{n+1}$	hard [44]	hard [51, 66]
Onto-PHP_n^{n+1}	hard [44]	hard [2]
Onto-$FPHP_n^{n+1}$	hard [44]	easy [59]

while the formula that contains all axioms (7.1a)-(7.1d) is called the *onto functional pigeonhole principle Onto-*$FPHP_n^{n+1}$. The overview of how these different versions of pigeonhole principle compare in resolution and polynomial calculus can be found in Table 7.1.

For resolution, Haken's celebrated result [44] established that pigeonhole principle is exponentially hard in terms of the number of holes. Moreover, it can be seen that this proof works for all other versions of the pigeonhole principle as well. On the other hand, in polynomial calculus it was known that the ordinary pigeonhole principle is hard by the result of Alekhnovich and Razborov [2], while the full onto functional pigeonhole principle had polynomially sized refutations as proved by Riis [59]. In the paper presented in this chapter, we have observed that Alekhnovich and Razborov's original proof extends to the onto pigeonhole principle, as well as used our framework to establish the exponential lower bound for the functional pigeonhole principle [51]. A similar lower bound for the functional pigeonhole principle, but proved directly without using any general framework, was also obtained by Wołochowski [66].

Chapter 8

Conclusion

In this thesis we explored the relation between space and width/degree in resolution and polynomial calculus, as well as different techniques for width/degree and therefore length/size lower bounds in these proof systems. In the first two papers of the thesis, Chapters 4 and 5, we explored the space lower bounds and the relation between space and width/degree. Building on previous results, we made progress on the question of whether degree is a lower bound on space in polynomial calculus, as well as proved that space cannot be a lower bound on degree. However, both of these results could be further improved. We still do not know whether degree remains a lower bound for space if we do not amplify the hardness of the formula by XORification. Furthermore, our second result where we separate degree from space in polynomial calculus depends on the characteristic of the field. That is, we need different formulas for different characteristics. We still do not know whether there are single formulas that separate degree from space for all fields simultaneously.

In studying the relation between space and degree in polynomial calculus, we have also explored this relation in resolution. We simplified the proof that width lower bounds space in resolution by directly transforming a small space refutation into a small width one. While the new proof helps us understand better the resolution result, it does not seem to help in any way with polynomial calculus. Hence, space in polynomial calculus is still only partially understood and there is ample room for improvement in the techniques for proving space lower bounds in order to truly capture all the formulas that we are interested in. In addition, improving these techniques could lead to resolving the previous question of whether space is lower bounded by degree, or even resolution width.

The other two papers in the thesis, Chapters 6 and 7, leave the topic of space and move to length/size lower bounds in resolution and polynomial calculus. By the result presented in Chapter 3, we have that width/degree lower bounds imply length/size lower bounds. In addition to exploring the question of length/size lower bounds, in Chapter 6 we also explore the practical question of how our theoretical lower bounds relate to actual SAT solver running times. We show the first theoretical

lower bound for subset cardinality formulas, which were previously shown to be among the hardest formulas in practice. In addition, we ran experiments that confirm these prior observations. One interesting result that we got in our experiments is that the theoretically easiest formulas turn out to be very hard in practice when we do not help the SAT solver by giving it an explicit ordering on the variables. This demonstrates that there are still open questions with regard to the relation between resolution and current state-of-the-art SAT solvers. Some progress on these questions was recently made by Elffers et al. [32].

In the final paper we turn to presenting a general framework for proving polynomial calculus degree lower bounds. We show that if we can form a special kind of a bipartite graph from a given CNF formula, then the degree lower bound follows. This method allows us to reprove almost all previously known degree lower bounds, as well as prove a lower bound for the functional pigeonhole principle. When we compare our framework to resolution, we can see that they are quite similar except that in resolution we have a weaker condition on the edges of the graph. However, this leads to a big difference as the resolution lower bound technique is applicable too almost all formulas that we care about. This is not the case with our polynomial calculus framework, as for instance we do not know how to fit coloring and independent set formulas into our framework although they are hard for resolution [6, 7]. Another open question is to find lower bounds for polynomial calculus size that do not go through degree lower bounds.

In conclusion, in this thesis we have made progress in understanding complexity measures in polynomial calculus. Generally, we can see that many general techniques and basic results transfer from resolution to polynomial calculus, but they can become quite harder to prove. This still leaves us with a lot of open questions in polynomial calculus and with the hope that we can find a way to simplify and unify these techniques. Achieving this goal could potentially not only help us understand resolution and polynomial calculus better, but might also provide us with tools to efficiently tackle even the more powerful proof systems.

Bibliography

[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] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. *Proceedings of the Steklov Institute of Mathematics*, 242:18–35, 2003. Available at `http://people.cs.uchicago.edu/~razborov/files/misha.pdf`. Preliminary version in *FOCS '01*.

[3] 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*.

[4] Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pages 203–208, July 1997.

[5] Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 213–232, May 2012.

[6] Paul Beame, Joseph C. Culberson, David G. Mitchell, and Cristopher Moore. The resolution complexity of random graph k-colorability. *Discrete Applied Mathematics*, 153(1-3):25–47, December 2005.

[7] Paul Beame, Russell Impagliazzo, and Ashish Sabharwal. The resolution complexity of independent sets and vertex covers in random graphs. *Computational Complexity*, 16(3):245–297, October 2007.

[8] Chris Beck and Russell Impagliazzo. Strong ETH holds for regular resolution. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 487–494, May 2013.

[9] Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.

[10] Eli Ben-Sasson. Size-space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, May 2009. Preliminary version in *STOC '02*.

[11] 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*.

[12] 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)*, pages 709–718, October 2008.

[13] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, January 2011.

[14] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

[15] Patrick Bennett, Ilario Bonacina, Nicola Galesi, Tony Huynh, Mike Molloy, and Paul Wollan. Space proof complexity for random 3-CNFs. Technical Report 1503.01613, arXiv.org, April 2015.

[16] Christoph Berkholz and Jakob Nordström. Supercritical space-width trade-offs for resolution. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP '16)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 57:1–57:14, July 2016.

[17] Armin Biere, Daniel Le Berre, Emmanuel Lonca, and Norbert Manthey. Detecting cardinality constraints in CNF. In *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *Lecture Notes in Computer Science*, pages 285–301. Springer, July 2014.

[18] Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.

[19] 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)*, pages 56:1–56:13, July 2016.

[20] Ilario Bonacina and Nicola Galesi. Pseudo-partitions, transversality and locality: A combinatorial characterization for the space measure in algebraic proof systems. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS '13)*, pages 455–472, January 2013.

[21] 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*.

[22] Ilario Bonacina and Navid Talebanfard. Improving resolution width lower bounds for k-cnfs with applications to the strong exponential time hypothesis. *Information Processing Letters*, 116:120–124, 2016.

[23] María Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, December 2001. Preliminary version in *FOCS '99*.

[24] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *Journal of Computer and System Sciences*, 62(2):267–289, March 2001. Preliminary version in *CCC '99*.

[25] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[26] 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)*, pages 174–183, May 1996.

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

[28] Stefan S. Dantchev and Søren Riis. "Planar" tautologies hard for resolution. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '01)*, pages 220–229, oct 2001.

[29] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, July 1962.

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

[31] Jan Elffers. Personal communication, 2015.

[32] Jan Elffers, Jan Johannsen, Massimo Lauria, Thomas Magnard, Jakob Nordström, and Marc Vinyals. Trade-offs between time and memory in a tighter model of CDCL SAT solvers. In *Proceedings of the 19th International Conference on Theory and Applications of Satisfiability Testing (SAT '16)*, volume 9710 of *Lecture Notes in Computer Science*, pages 160–176. Springer, July 2016.

[33] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS '99)*, volume 1563 of *Lecture Notes in Computer Science*, pages 551–560. Springer, 1999.

[34] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Based on the conference papers in *STACS '99* [33] and *CSL '99* [63].

[35] Yuval Filmus. On the Alekhnovich–Razborov degree lower bound. Manuscript. Available at `http://www.cs.toronto.edu/~yuvalf/AlRa.pdf`, October 2014.

[36] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (Extended abstract). In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 437–448. Springer, July 2013.

[37] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. From small space to small width in resolution. In *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS '14)*, volume 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 300–311, March 2014.

[38] 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*.

[39] 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*.

[40] Nicola Galesi and Massimo Lauria. On the automatizability of polynomial calculus. *Theory of Computing Systems*, 47:491–506, August 2010.

[41] Nicola Galesi and Massimo Lauria. Optimality of size-degree trade-offs for polynomial calculus. *ACM Transactions on Computational Logic*, 12:4:1–4:22, November 2010.

[42] The Glucose SAT solver. `http://www.labri.fr/perso/lsimon/glucose/`.

[43] Dima Grigoriev. Tseitin's tautologies and lower bounds for Nullstellensatz proofs. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS '98)*, pages 648–652, November 1998.

[44] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[45] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity (Extended abstract). In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 233–248, May 2012.

[46] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[47] Lingeling and Plingeling. `http://fmv.jku.at/lingeling/`.

[48] March. `http://www.st.ewi.tudelft.nl/~marijn/sat/march\_dl.php`.

[49] João P. Marques-Silva and Karem A. Sakallah. GRASP—a new search algorithm for satisfiability. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '96)*, pages 220–227, November 1996.

[50] Mladen Mikša and Jakob Nordström. Long proofs of (seemingly) simple formulas. In *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *Lecture Notes in Computer Science*, pages 121–137. Springer, July 2014.

[51] Mladen Mikša and Jakob Nordström. A generalized method for proving polynomial calculus degree lower bounds. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 467–487, June 2015.

[52] The MiniSat page. `http://minisat.se/`.

[53] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC '01)*, pages 530–535, June 2001.

[54] Jakob Nordström. A simplified way of proving trade-off results for resolution. *Information Processing Letters*, 109(18):1030–1035, August 2009. Preliminary version in ECCC report TR07-114, 2007.

[55] Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

[56] Alexander Razborov. Personal communication, 2014.

[57] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, December 1998.

[58] Alexander A. Razborov. Proof complexity of pigeonhole principles. In *5th International Conference on Developments in Language Theory, (DLT '01), Revised Papers*, volume 2295 of *Lecture Notes in Computer Science*, pages 100–116. Springer, July 2002.

[59] Søren Riis. *Independence in Bounded Arithmetic*. PhD thesis, University of Oxford, 1993.

[60] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[61] Michael Sipser. *Introduction to the Theory of Computation*. Cengage Learning, 3rd edition, 2012.

[62] Ivor Spence. sgen1: A generator of small but difficult satisfiability benchmarks. *Journal of Experimental Algorithmics*, 15:1.2:1.1–1.2:1.15, March 2010.

[63] Jacobo Torán. Lower bounds for space in resolution. In *Proceedings of the 13th International Workshop on Computer Science Logic (CSL '99)*, volume 1683 of *Lecture Notes in Computer Science*, pages 362–373. Springer, 1999.

[64] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1): 209–219, January 1987.

[65] Allen Van Gelder and Ivor Spence. Zero-one designs produce small hard SAT instances. In *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT '10)*, volume 6175 of *Lecture Notes in Computer Science*, pages 388–397. Springer, July 2010.

[66] Lukasz Wołochowski. An application of expanders to the study of proofs of the pigeonhole principle in algebraic propositional proof systems. Master's thesis, University of Warsaw, 2013.

Part II

Publications

A

Towards an Understanding of Polynomial Calculus: New Separations and Lower Bounds*

Yuval Filmus[1], Massimo Lauria[2], Mladen Mikša[2], Jakob Nordström[2], and Marc Vinyals[2]

[1]University of Toronto
[2]KTH Royal Institute of Technology

Abstract

During the last decade, an active line of research in proof complexity has been into the space complexity of proofs and how space is related to other measures. By now these aspects of resolution are fairly well understood, but many open problems remain for the related but stronger polynomial calculus (PC/PCR) proof system. For instance, the space complexity of many standard "benchmark formulas" is still open, as well as the relation of space to size and degree in PC/PCR.

We prove that if a formula requires large resolution width, then making XOR substitution yields a formula requiring large PCR space, providing some circumstantial evidence that degree might be a lower bound for space. More importantly, this immediately yields formulas that are very hard for space but very easy for size, exhibiting a size-space separation similar to what is known for resolution. Using related ideas, we show that if a graph has good expansion and in addition its edge set can be partitioned into short cycles, then the Tseitin formula over this graph requires large PCR space. In particular, Tseitin formulas over random 4-regular graphs almost surely require space at least $\Omega\big(\sqrt{n}\big)$.

Our proofs use techniques recently introduced in [Bonacina-Galesi '13]. Our final contribution, however, is to show that these techniques provably cannot yield non-constant space lower bounds for the functional pigeonhole principle, delineating the limitations of this framework and suggesting that we are still far from characterizing PC/PCR space.

*This is the full-length version of the paper [FLM+13] that appeared in *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13).*

1 Introduction

Proof complexity studies how hard it is to provide succinct certificates for tautological formulas in propositional logic—i.e., proofs that formulas always evaluate to true under any truth value assignment, where these proofs are verifiable in time polynomial in their size. It is widely believed that there is no proof system where such efficiently verifiable proofs can always be found of size at most polynomial in the size of the formulas they prove. Showing this would establish $\mathsf{NP} \neq \mathsf{coNP}$, and hence $\mathsf{P} \neq \mathsf{NP}$, and the study of proof complexity was initiated by Cook and Reckhow [CR79] as an approach towards this (still very distant) goal.

A second prominent motivation for proof complexity is the connection to applied SAT solving. By a standard transformation, any propositional logic formula F can be transformed to another formula F' in conjunctive normal form (CNF) such that F' has the same size up to constant factors and is unsatisfiable if and only if F is a tautology. Any algorithm for solving SAT defines a proof system in the sense that the execution trace of the algorithm constitutes a polynomial-time verifiable witness of unsatisfiability (such a witness is often referred to as a *refutation* rather than a *proof*, and we will use the two terms interchangeably in this paper). In the other direction, most modern SAT solvers can in fact be seen to search for proofs in systems studied in proof complexity, and upper and lower bounds for these proof systems hence give information about the potential and limitations of such SAT solvers.

In addition to running time, a major concern in SAT solving is memory consumption. In proof complexity, these two resources are modelled by *proof size/length* and *proof space*. It is thus interesting to understand these complexity measures and how they are related to each other, and such a study reveals intriguing connections that are also of intrinsic interest to proof complexity. In this context, it is natural to focus on proof systems at comparatively low levels in the proof complexity hierarchy that are, or could plausibly be, used as a basis for SAT solvers. Such proof systems include resolution and polynomial calculus. This paper takes as its starting point the former system but focuses on the latter.

Previous Work

The *resolution* proof system was introduced in [Bla37], and is at the foundation of state-of-the-art SAT solvers based on so-called conflict-driven clause learning (CDCL) [BS97, MS96].

In resolution, one derives new disjunctive clauses from the clauses of the original CNF formula until contradiction is reached. One of the early breakthroughs in proof complexity was the (sub)exponential lower bound on proof length (measured as the number of clauses in a proof) obtained by Haken [Hak85]. Truly exponential lower bounds—i.e., bounds $\exp(\Omega(n))$ in the size n of the formula—were later established in [CS88, Urq87] and other papers.

Ben-Sasson and Wigderson [BW01] identified *width* as a crucial resource, where the width is the size of a largest clause in a resolution proof. They proved that strong lower bounds on width imply strong lower bounds on length, and used this to rederive essentially all known length lower bounds in terms of width.

The study of space in resolution was initiated by Esteban and Torán [ET01], measuring the space of a proof (informally) as the maximum number of clauses needed to be kept in memory during proof verification. Alekhnovich et al. [ABRW02] later extended the concept of space to a more general setting, including other proof systems. The (clause) space measure can be shown to be at most linear in the formula size, and matching lower bounds were proven in [ABRW02, BG03, ET01].

Atserias and Dalmau [AD08] proved that space is in fact lower-bounded by width, which allowed to rederive all hitherto known space lower bounds as corollaries of width lower bounds. A strong separation of the two measures was obtained in [BN08], exhibiting a formula family with constant width complexity but almost linear space complexity. Also, dramatic space-width trade-offs have been shown in [Ben09], with formulas refutable in constant width and constant space where optimizing one of the measures causes essentially worst-case behaviour of the other.

Regarding the connections between length and space, it follows from [AD08] that formulas of low space complexity also have short proofs. For the subsystem of *tree-like resolution*, where each line in the proof can only be used once, [ET01] showed that length upper bounds also imply space upper bounds, but for general resolution [BN08] established that this is false in the strongest possible sense. Strong trade-offs between length and space were proven in [BN11, BBI12].

This paper focuses on the more powerful *polynomial calculus (PC)*[1] proof system introduced by Clegg et al. [CEI96], which is not at all as well understood. In a PC proof, clauses are interpreted as multilinear polynomials (expanded out to sums of monomials), and one derives contradiction by showing that these polynomials have no common root. Intriguingly, while proof complexity-theoretic results seem to hold out the promise that SAT solvers based on PC could be orders of magnitude faster than CDCL, such algebraic solvers have so far failed to be truly competitive.

Proof size[2] in PC is measured as the total number of monomials in a proof and the analogue of resolution space is the number of monomials needed in memory during proof verification. Clause width in resolution translates into polynomial degree in PC. While length, space and width in resolution are fairly well understood as surveyed above, our understanding of the corresponding complexity measures in PC is much more limited.

Impagliazzo et al. [IPS99] showed that strong degree lower bounds imply strong size lower bounds. This is a parallel to the length-width relation in [BW01], and in

[1] Strictly speaking, to get a stronger proof system than resolution we need to look at the generalization *PCR* as defined in [ABRW02], but for simplicity we will be somewhat sloppy in this introduction in distinguishing between PC and PCR.

[2] The *length* of a proof is the number of lines, whereas *size* also considers the size of lines. In resolution the two measures are essentially equivalent. In PC size and length can be very different, however, and so size is the right measure to study.

fact this latter paper can be seen as a translation of the bound in [IPS99] from PC to resolution. This size-degree relation has been used to prove exponential lower bounds on size in a number of papers, with [AR03] perhaps providing the most general setting.

The first lower bounds on space were reported in [ABRW02], but only sublinear bounds and only for formulas of unbounded width. The first space lower bounds for k-CNF formulas were presented in [FLN$^+$12], and asymptotically optimal (linear) lower bounds were finally proven by Bonacina and Galesi [BG13]. However, there are several formula families with high resolution space complexity for which the PC space complexity has remained unknown, e.g., Tseitin formulas (encoding that the sum of all vertex degrees in an undirected graph must be even), ordering principle formulas, and functional pigeonhole principle (FPHP) formulas.

Regarding the relation between space and degree, it is open whether degree is a lower bound for space (which would be the analogue of what holds in resolution) and also it has been unknown whether the two measures can be separated in the sense that there are formulas of low degree complexity requiring high space. However, [BNT13] recently proved a space-degree trade-off analogous to the resolution space-width trade-off in [Ben09] (in fact for the very same formulas). This could be interpreted as indicating that there should be a space-degree separation analogous to the space-width separation in resolution, and the authors of [BG13] suggest that their techniques might be a step towards understanding degree and proving that degree lower-bounds space, similar to how this was done for resolution width in [AD08].

As to size versus space in PC, essentially nothing has been known. It is open whether small space complexity implies small size complexity and/or the other way around. Some size-space trade-offs were recently reported in [HN12, BNT13], but these trade-offs are weaker than the corresponding results for resolution.

Our Results

We study the relation of size, space, and degree in PC (and the stronger system PCR) and present a number of new results as briefly described below.

1. We prove that if the resolution width of refuting a CNF formula F is w, then by substituting each variable by an exclusive or of two new variables and expanding out we get a new CNF formula $F[\oplus]$ requiring PCR space $\Omega(w)$. In one sense, this is stronger than claiming that degree is a lower bound for space, since high width complexity is a necessary but not sufficient condition for high degree complexity. In another sense, however, this is (much) weaker in that XOR substitution can amplify the hardness of formulas substantially. Nevertheless, to the best of our knowledge this is the first result making any connection between width/degree and space for polynomial calculus.

2. More importantly, this result yields essentially optimal separations between length and degree on the one hand and space on the other. Namely, taking

expander graphs and making double copies of all edges, we show that Tseitin formulas over such graphs have proofs in size $O(n \log n)$ and degree $O(1)$ in PC but require space $\Theta(n)$ in PCR. (Furthermore, since these small-size proofs are tree-like, this shows that there is no tight correlation between size and space in tree-like PC/PCR in contrast to resolution.)

3. Using related ideas, we also prove strong PCR space lower bounds for Tseitin formulas over (simple or multi-)graphs where the edge set can be partitioned into small cycles. (The two copies of every edge in the multi-graph above form such cycles, but this works in greater generality.) In particular, for Tseitin formulas over random d-regular graphs for $d \geq 4$ we establish that an $\Omega(\sqrt{n})$ PCR space lower bound holds asymptotically almost surely.

4. On the negative side, we show that the techniques in [BG13] cannot prove any non-constant PCR space lower bounds for functional pigeonhole principle (FPHP) formulas. That is, although these formulas require high degree and it seems plausible that they are hard also with respect to space, the machinery developed in [BG13] provably cannot establish such lower bounds. Unfortunately, this seems to indicate that we are further from characterizing degree in PC/PCR than previously hoped.

Organization of This Paper

The rest of this paper is organized as follows. We briefly review preliminaries in Section 2. Section 3 presents a overview of our results and provides some proof sketches outlining the main technical ideas that go into the proofs.

In Section 4, we prove that resolution width lower bounds plus substitutions with XOR or other suitable Boolean functions yields PCR space lower bounds. We use this in Section 5 to separate size and degree from space in PC and PCR. In Section 6, we show PCR space lower bounds for Tseitin formulas over graphs with edge sets decomposable into partitions of small cycles. The proof that random d-regular graphs for $d \geq 4$ (almost) decompose into cycles of length $O(\sqrt{n})$ is given in Section 7. The fact that PCR space lower bounds cannot be obtained for the functional pigeonhole principle formulas with current techniques is proven in Section 8, and in the same section we show that a larger class of formulas containing FPHP formulas have essentially the same space complexity for PC and PCR (so that when proving lower bounds, one can without loss of generality ignore the complementary formal variables for negative literals in PCR and focus only on PC).

We make some concluding remarks and discuss some of the (many) open questions remaining in Section 9. For completeness, in Appendix A we provide a full description of our version of the techniques in [BG13] and provide proofs that the same claims still hold in this slightly different setting.

2 Preliminaries

A *literal* over a Boolean variable x is either the variable x itself (a *positive literal*) or its negation $\neg x$ or $\overline{x}$ (a *negative literal*). It will also be convenient to use the alternative notation $x^0 = x$, $x^1 = \overline{x}$, where we identify 0 with true and 1 with false[3] (so that x^b is true if $x = b$). A *clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of literals. We denote the empty clause by $\bot$. A clause containing at most k literals is called a *k-clause*. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses. A *k-CNF formula* is a CNF formula consisting of k-clauses. We think of clauses and CNF formulas as sets so that order is irrelevant and there is no repetitions.

Let $\mathbb{F}$ be a field and consider the polynomial ring $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$ (where x and $\overline{x}$ are viewed as distinct formal variables). We employ the standard notation $[n] = \{1, \ldots, n\}$.

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

Axiom download $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is either a monomial $m = \prod_i x_i^b$ encoding a clause $C = \bigvee_i x_i^b \in F$, or a *Boolean axiom* $x^2 - x$ or *complementarity axiom* $x + \overline{x} - 1$ for any variable x (or $\overline{x}$).

Inference $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{p\}$, where p is inferred by *linear combination* $\frac{q \quad r}{\alpha q + \beta r}$ or *multiplication* $\frac{q}{xq}$ from polynomials $q, r \in \mathbb{P}_{t-1}$ for $\alpha, \beta \in \mathbb{F}$ and x a variable.

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

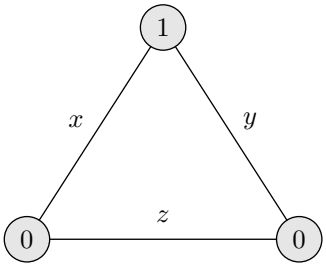
If we drop complementarity axioms and encode each negative literal $\overline{x}$ as the polynomial $(1 - x)$, the proof system is called *polynomial calculus (PC)*.

The *size* $S(\pi)$ of a PC/PCR refutation π is the number of monomials (counted with repetitions) in all downloaded or derived polynomials in π, the *(monomial) space* $Sp(\pi)$ is the maximal number of monomials (counted with repetitions)[4] in any configuration in π, and the *degree* $Deg(\pi)$ is the maximal degree of any monomial appearing in π. Taking the minimum over all PCR refutations of a formula F, we define the size $S_{\mathcal{PCR}}(F \vdash \bot)$, space $Sp_{\mathcal{PCR}}(F \vdash \bot)$, and degree $Deg_{\mathcal{PCR}}(F \vdash \bot)$ of refuting F in PCR (and analogously for PC).

We can also define *resolution* in this framework, where proof lines are always clauses (i.e., single monomials) and new clauses can be derived by the *resolution rule* inferring $C \vee D$ from $C \vee x$ and $D \vee \overline{x}$. The *length* of a resolution refutation π is the number of downloaded and derived clauses, the *space* is the maximal number of

[3] Note that this notational convention is the opposite of what is found in many other papers, but as we will see shortly it is the natural choice in the context of polynomial calculus.

[4] We note that in [ABRW02], space was defined *without* counting repetitions of monomials. All our lower bounds hold in this more stringent setting as well.



(a) Labelled triangle graph.

$$
\begin{aligned}
&(x \vee y)\\
\wedge\ &(\overline{x} \vee \overline{y})\\
\wedge\ &(x \vee \overline{z})\\
\wedge\ &(\overline{x} \vee z)\\
\wedge\ &(y \vee \overline{z})\\
\wedge\ &(\overline{y} \vee z)
\end{aligned}
$$

(b) Corresponding Tseitin formula.

Figure 2: Example Tseitin formula.

clauses in any configuration, and the *width* is the size of a largest clause appearing in π (or equivalently the degree of such a monomial). Taking the minimum over all refutations as above we get the measures $L_{\mathcal{R}}(F \vdash \bot)$, $Sp_{\mathcal{R}}(F \vdash \bot)$, and $W_{\mathcal{R}}(F \vdash \bot)$. It is not hard to show that PCR can simulate resolution efficiently with respect to all these measures.

We say that a refutation is *tree-like* if every line is used at most once as the premise of an inference rule before being erased (though it can possibly be rederived later). All measures discussed above can also be defined for restricted subsystems of resolution, PC and PCR admitting only tree-like refutations.

Let us now describe the family of CNF formulas which will be the main focus of our study.

Definition 2 (Tseitin formula). Let $G = (V, E)$ be an undirected graph and $\chi : V \to \{0, 1\}$ be a function. Identify every edge $e \in E$ with a variable x_e and let $PARITY_{v,\chi}$ denote the CNF encoding of the constraint that the number of true edges x_e incident to a vertex $v \in V$ is equal to $\chi(v) \pmod 2$. Then the *Tseitin formula* over G with respect to χ is $Ts(G, \chi) = \bigwedge_{v \in V} PARITY_{v,\chi}$.

When the degree of G is bounded by d, $PARITY_{v,\chi}$ has at most 2^{d-1} clauses, all of width at most d, and hence $Ts(G, \chi)$ is a d-CNF formula with at most $2^{d-1}|V|$ clauses. Figure 1(b) gives an example Tseitin formula generated from the graph in Figure 1(a). We say that a set of vertices U has *odd (even) charge* if $\sum_{u \in U} \chi(u)$ is odd (even). By a simple counting argument one sees that $Ts(G, \chi)$ is unsatisfiable if $V(G)$ has odd charge. Lower bounds on the hardness of refuting such unsatisfiable formulas $Ts(G, \chi)$ can be proven in terms of the expansion of G as defined next.

Definition 3 (Connectivity expansion [ABRW02]). The *connectivity expansion* of $G = (V, E)$ is the largest c such that for every $E' \subseteq E$, $|E'| \leq c$, the graph $G' = (V, E \setminus E')$ has a connected component of size strictly greater than $|V|/2$.

If F is a CNF formula and $f : \{0,1\}^d \to \{0,1\}$ is a Boolean function, then we can obtain a new CNF formula by substituting $f(x_1, \ldots, x_d)$ for every variable x and expanding out to conjunctive normal form. We write $F[f]$ to denote the resulting *substituted formula*, where we will be interested in substitutions with a particular kind of functions defined as follows.

Definition 4 (Non-authoritarian function [BN11]). We say that a Boolean function $f(x_1, \ldots, x_d)$ is *non-authoritarian* if for every x_i and for every assignment α to x_i there exist α_0, α_1 extending α such that $f(\alpha_b) = b$ for $b \in \{0,1\}$.

By way of example, exclusive or (XOR), denoted $\oplus$, is clearly non-authoritarian, since regardless of the value of one variable, the other one can be flipped to make the function true or false, but standard non-exclusive or $\vee$ is not.

Let us finally give a brief overview of the framework developed in [BG13], which we use to prove our PCR space lower bounds.[5] A *partial partition* $\mathcal{Q}$ of a variable set V is a collection of disjoint sets $Q_i \subseteq V$. We use the notation $\bigcup \mathcal{Q} = \bigcup_{Q_i \in \mathcal{Q}} Q_i$. For two sets of partial assignments H and H' to disjoint domains, we denote by $H \times H'$ the set of assignments $H \times H' = \{\alpha \cup \beta \mid \alpha \in H \text{ and } \beta \in H'\}$. A set of partial assignments H to the domain Q is *flippable* on Q if for each variable $x \in Q$ and $b \in \{0,1\}$ there exists an assignment $\alpha_b \in H$ such that $\alpha_b(x) = b$. We say that H *satisfies* a formula F if all $\alpha \in H$ satisfy F.

A $\mathcal{Q}$*-structured assignment set* is a pair $(\mathcal{Q}, \mathcal{H})$ consisting of a partial partition $\mathcal{Q} = \{Q_1, \ldots, Q_t\}$ of V and a set of partial assignments $\mathcal{H} = \prod_{i=1}^{t} H_i$, where each H_i assigns to and is flippable on Q_i. We write $(\mathcal{Q}, \mathcal{H}) \preccurlyeq (\mathcal{Q}', \mathcal{H}')$ if $\mathcal{Q} \subseteq \mathcal{Q}'$ and $\mathcal{H}' \restriction_{\mathcal{Q}} = \mathcal{H}$, where $\mathcal{H}' \restriction_{\mathcal{Q}} = \prod_{Q_i \in \mathcal{Q}} H'_i$. A structured assignment set $(\mathcal{Q}, \mathcal{H})$ *respects* a CNF formula F' if for every clause $C \in F'$ either $\mathit{Vars}(C) \cap \bigcup \mathcal{Q} = \emptyset$ or there is a set $Q \in \mathcal{Q}$ such that $\mathit{Vars}(C) \subseteq Q$ and $\mathcal{H}$ satisfies C.

Expressed in this language, the key technical definition in [BG13] is as follows.

Definition 5 (Extendible family). A non-empty family $\mathcal{F}$ of structured assignment sets $(\mathcal{Q}, \mathcal{H})$ is r*-extendible* for a CNF formula F with respect to a satisfiable $F' \subseteq F$ if every $(\mathcal{Q}, \mathcal{H}) \in \mathcal{F}$ satisfies the following conditions.

Size $|\mathcal{Q}| \leq r$.

Respectfulness $(\mathcal{Q}, \mathcal{H})$ respects F'.

Restrictability For every $\mathcal{Q}' \subseteq \mathcal{Q}$ the restriction $(\mathcal{Q}', \mathcal{H} \restriction_{\mathcal{Q}'})$ is in $\mathcal{F}$.

Extendibility If $|\mathcal{Q}| < r$ then for every clause $C \in F \setminus F'$ there exists $(\mathcal{Q}', \mathcal{H}') \in \mathcal{F}$ such that 1. $(\mathcal{Q}, \mathcal{H}) \preccurlyeq (\mathcal{Q}', \mathcal{H}')$, 2. $\mathcal{H}'$ satisfies C, and 3. $|\mathcal{Q}'| \leq |\mathcal{Q}| + 1$.

When $F' = \emptyset$, we simply say that $\mathcal{F}$ is r-extendible for F.

[5] The actual definitions that we use are slightly different but essentially equivalent. We provide the full details including proofs in Section A for completeness.

To prove PCR space lower bounds for a formula F, it is sufficient to find an extendible family for F.

Theorem 6 ([BG13]). *Suppose that F is a CNF formula which has an r-extendible family $\mathcal{F}$ with respect to some $F' \subseteq F$. Then $Sp_{\mathcal{PCR}}(F \vdash \bot) \geq r/4$.*

All space lower bounds presented in this paper are obtained in this manner, where in addition we always have $F' = \emptyset$.

3 Overview of Results and Sketches of Some Proofs

In this section, we give a more detailed overview with formal statements of our results, and also provide some proof sketches in order to convey the main technical ideas. As a general rule, the upper bounds we state are for polynomial calculus (PC) whereas the lower bounds hold for the stronger system polynomial calculus resolution (PCR). In fact, even more can be said: just as is the case in [ABRW02, FLN$^+$12, BG13], all our lower bounds hold also for *functional calculus*, where proof lines are arbitrary Boolean functions over clauses/monomials and anything that follows semantically from the current configuration can be derived in a single step. We do not discuss this further below but instead refer to Appendix A for the details.

Relating PCR Space and Resolution Width

The starting point of our work is the question of how space and degree are related in polynomial calculus, and in particular whether it is true that degree lower-bounds space. While this question remains wide open, we make partial progress by showing that if the resolution width of refuting a CNF formula F is large (which in particular must be the case if F requires high degree), then by making XOR substitution we obtain a formula $F[\oplus]$ that requires large PCR space. In fact, this works not only for exclusive or but for any non-authoritarian function (as defined in Definition 4). The formal statement is as follows.

Theorem 7. *Let F be a k-CNF formula and let f be any non-authoritarian function. Then it holds over any field that $Sp_{\mathcal{PCR}}(F[f] \vdash \bot) \geq (W_{\mathcal{R}}(F \vdash \bot) - k + 1)/4$.*

Proof sketch. In one sentence, the proof of Theorem 7 is by combining the concept of extendible families in Definition 5 with the combinatorial characterization of resolution width in [AD08]. We show that the properties of F implied by the width lower bound can be used to construct an extendible family for $F[f]$. To make this description easier to parse, let us start by describing in somewhat more detail the width characterization in [AD08].

Consider the following game played on F by two players *Spoiler* and *Duplicator*. Spoiler asks about assignments to variables in F and Duplicator answers true or false. Spoiler can only remember ℓ assignments simultaneously, however, and has to forget some variable when this limit is reached. If Duplicator is later asked

about some forgotten variable, the new assignment need not be consistent with the previous forgotten one. Spoiler wins the game by constructing a partial assignment that falsifies some clause in F, and the game is a Duplicator win if there is a strategy to keep playing forever without Spoiler ever reaching this goal. It was proven in [AD08] that this game exactly captures resolution width in the sense that Duplicator has a winning strategy if and only if $\ell \leq W_{\mathcal{R}}(F \vdash \bot)$.

Let us fix $r = W_{\mathcal{R}}(F \vdash \bot) - k + 1$ and use Duplicator's winning strategy for $\ell = W_{\mathcal{R}}(F \vdash \bot)$ to build an r-extendible family for $F[\oplus]$ (the proof for general non-authoritarian functions is very similar and is given in Section 4). Consider any assignment α reached during the game. We define a corresponding structured assignment set $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha)$ by adding a block $Q_x = \{x_1, x_2\}$ to $\mathcal{Q}_\alpha$ for every $x \in \mathrm{Dom}(\alpha)$, and let H_x contain all assignments α_x to $\{x_1, x_2\}$ such that $\alpha_x(x_1 \oplus x_2) = \alpha(x)$.

Given these structured assignment sets $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha)$, the family $\mathcal{F}$ is constructed inductively as follows. The base case is that $(\mathcal{Q}_\emptyset, \mathcal{H}_\emptyset) = (\emptyset, \emptyset)$ is in $\mathcal{F}$. To extend $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha)$ to satisfy a clause in $C[\oplus]$, we simulate a Spoiler with memory α who asks about all variables in C. Since Duplicator does not falsify C, when all variables have been queried some literal in C must be satisfied by the assignment. Fix one such variable assignment $\{x = b\}$ and add $\left(\mathcal{Q}_{\alpha \cup \{x=b\}}, \mathcal{H}_{\alpha \cup \{x=b\}}\right)$ as defined above to $\mathcal{F}$. All that remains now is to verify that this yields an extendible family as described in Definition 5 and then apply Theorem 6. □

Separation of Size and Degree from Space

An almost immediate consequence of Theorem 7 is that there are formulas which have small PC refutations in constant degree but nevertheless require maximal space in PCR.

Theorem 8. *For any field $\mathbb{F}$ of characteristic p there is a family of k-CNF formulas F_n (where k depends on p) of size $\mathrm{O}(n)$ for which $Sp_{\mathcal{PCR}}(F_n \vdash \bot) = \Omega(n)$ over any field but which have tree-like PC refutations $\pi_n : F_n \vdash \bot$ over $\mathbb{F}$ of size $S(\pi_n) = \mathrm{O}(n \log n)$ and degree $Deg(\pi_n) = \mathrm{O}(1)$.*

Proof sketch. Let us focus on $p = 2$, deferring the general proof to Section 5. Consider a Tseitin formula $Ts(G, \chi)$ for any constant-degree graph G over n vertices with connectivity expansion $\Omega(n)$ and any odd-charge function χ.

From [BW01] we know that $W_{\mathcal{R}}(F \vdash \bot) = \Omega(n)$. It is not hard to see that XOR substitution yields another Tseitin formula $Ts(G', \chi)$ for the multi-graph G' obtained from G by adding double copies of all edges. This formula requires large PCR space (over any field) by Theorem 7. The upper bound follows by observing that the CNF encodes a linear system of equations, which is easily shown inconsistent in PC by summing up all equations in a tree-like fashion. □

It follows from Theorem 8 that tree-like space in PC/PCR is not upper-bounded by tree-like size, in contrast to resolution. This is the only example we are aware of where the relations between size, degree, and space in PC/PCR differ from those

between length, width, and space in resolution, so let us state this as a formal corollary.

Corollary 9. *It is* not true *in PC/PCR that tree-like space complexity is upper-bounded by the logarithm of tree-like size complexity.*

Space Complexity of Tseitin Formulas

A closer analysis of the proof of Theorem 8 reveals that it partitions the edge set of G' into small edge-disjoint cycles (namely, length-2 cycles corresponding to the two copies of each original edge) and uses partial assignments that all maintain the same parities of the vertices on a given cycle. It turns out that this approach can be made to work in greater generality as stated next.

Theorem 10. *Let $G = (V, E)$ be a connected graph of bounded degree d with connectivity expansion c such that the edge set E can be partitioned into cycles of length at most b. Then it holds over any field that $Sp_{\mathcal{PCR}}(Ts(G, \chi) \vdash \bot) \geq c/4b - d/8$.*

Proof sketch. We build on the resolution space lower bound in [ABRW02, ET01], where the proof works by inductively constructing an assignment α_t for each derived configuration $\mathbb{C}_t$ (which corresponds to removing edges from G and updating the vertex charges accordingly) such that (a) α_t satisfies $\mathbb{C}_t$, and (b) α_t does not create any odd-charge component in G of size less than $n/2$. The inductive update can be performed as long as the space is not too large, which shows that contradiction cannot be derived in small space (since $\mathbb{C}_t$ is satisfiable).

To lift this proof to PCR, however, we must maintain not just one but an exponential number of such good assignments, and in general we do not know how to do this. Nevertheless, some more thought reveals that the only important aspect of our assignments are the resulting vertex parities. And if the edge set is partitioned into cycles, we can always shift edge charges along the cycles so that for all the exponentially many assignments, the vertex parities are all the same (meaning that on a higher level we only have to maintain one good assignment after all). The full proof is presented in Section 6. □

Some graphs, such as rectangular grids, can be partitioned into cycles of size $O(1)$, yielding tight bounds on space. A bit more surprisingly, random d-regular graphs for $d \geq 4$ turn out to (sort of) admit partitions into cycles of size $O(\sqrt{n})$, which yields the following theorem.

Theorem 11. *Let G be a random d-regular graph on n vertices, where $d \geq 4$. Then over any field it holds almost surely that $Sp_{\mathcal{PCR}}(Ts(G, \chi) \vdash \bot) = \Omega(\sqrt{n})$.*

Proof sketch. As long as we are interested in properties holding asymptotically almost surely, we can replace random 4-regular graphs with unions of two random Hamiltonian cycles [KW01]. We show that a graph distributed according to the latter model almost surely decomposes into cycles of length $O(\sqrt{n})$, along with εn

additional edges (which are easily taken care of separately). Since random graphs are also excellent expanders, we can apply Theorem 10. The argument extends straightforwardly to random d-regular graphs for any $d \geq 4$. The full proof, which contains a bit more by way of technical details, is given in Section 7. □

We believe that the true space bound should actually be $\Theta(n)$, just as for resolution, but such a result seems beyond the reach of our current techniques. Also, note that to make Theorem 10 go through we need graph expansion *plus* partitions into small cycles. It seems plausible that expansion alone should be enough to imply PCR space lower bounds, as for resolution, but again we are not able to prove this.

Limitations of the PCR Space Lower Bound Technique

The framework in [BG13] can also be used to rederive all PCR space lower bounds shown previously in [ABRW02, FLN$^+$12], and in this sense [BG13] sums up what we know about PCR space lower bounds. There are also intriguing similarities between [BG13] and the resolution width characterization in [AD08] (as partly hinted in the proof sketch for Theorem 7), which raises the question whether extendible families could perhaps be a step towards characterizing degree and showing that degree lower-bounds space in PC/PCR.

Even more intriguingly, however, there are CNF formulas for which it seems reasonable to expect that PCR space lower bounds should hold, but where extendible families seem very hard to construct. Such formulas include ordering principle formulas, functional pigeonhole principle (FPHP) formulas, and random 3-CNF formulas. In fact, no PCR space lower bounds are known for *any* 3-CNF formula—it is consistent with current knowledge that all 3-CNF formulas could have constant space complexity in PCR (!), though this seemingly absurd possibility can be ruled out for PC [FLN$^+$12].

We show that the problems in applying [BG13] to the functional version of the pigeonhole principle are inherent, in that these techniques provably cannot establish *any* nontrivial space lower bound. We refer to Section 8 for the formal description of the formulas and the proof of the next theorem.

Theorem 12. *There is no r-extendible family for $FPHP_n^{n+1}$ for $r > 1$.*

Since by [Raz98] these formulas[6] require PC refutation degree $\Omega(n)$, one way of interpreting Theorem 12 is that the concept of r-extendible families is very far from providing the hoped-for characterization of degree.

One step towards proving PCR space lower bounds could be to obtain a weaker PC space lower bound—as noted above in the discussion of 3-CNF formulas, this can sometimes be easier. For $FPHP_n^{n+1}$, however, and for a slightly more general

[6]To be precise, the degree lower bound in [Raz98] is proven for the functional pigeonhole principle encoded as linear equations—the standard CNF version has large initial width/degree and so there is nothing to prove. However, the linear-equations encoding of FPHP has axioms of large space, and so for space lower bounds we want to study the CNF version.

class of formulas described in Section 8, it turns out that such PC space lower bounds would immediately imply also PCR space lower bounds.

Theorem 13. $Sp_{\mathcal{PCR}}(FPHP_n^{n+1} \vdash \bot) = \Theta(Sp_{\mathcal{PC}}(FPHP_n^{n+1} \vdash \bot))$.

Proof sketch. In $FPHP_n^{n+1}$ we have variables $x_{i,j}$ for $i \in [n+1]$, $j \in [n]$, encoding that pigeon i goes into hole j. The clauses of the formula require that every pigeon is mapped to some hole and that this mapping is one-to-one. Because of this, the negation of $x_{i,j}$ is equivalent to $\bigvee_{j' \neq j} x_{i,j'}$ and so the literal $\overline{x}_{i,j}$ can be encoded as the monomial $\prod_{j' \neq j} x_{i,j'}$ in PC. Since this substitutes a monomial for a monomial the space does not increase. Now we can take any PCR refutation of $FPHP_n^{n+1}$ and apply such substitutions line by line. The inferences remain sound (with some local auxiliary steps added) and so this process gives a PC refutation of $FPHP_n^{n+1}$ in roughly the same space. □

4 PCR Space Lower Bounds From Resolution Width

In the rest of this paper, we give formal proofs of the results described in Section 3. We start by considering the question of relating space and degree in PCR. Although we do not know how to prove (or rule out) an analogue of the relation between space and width in resolution, we can use the combinatorial game from [AD08] to prove a weaker relation between PCR space and resolution width. Recall from the informal description of the game in Section 3 that we have two players, Spoiler and Duplicator, and that Duplicator needs to be able to provide an answer to any of Spoiler's questions about assignments to some bounded number of variables in order to win the game. Formally, a winning strategy for Duplicator is defined as follows.

Definition 14 (Duplicator's strategy [AD08]). A *Duplicator winning strategy* for the Boolean existential ℓ-pebble game on a CNF formula F is a non-empty family $\mathcal{D}$ of partial truth value assignments to $Vars(F)$ such that every $\alpha \in \mathcal{D}$ satisfies the following conditions:

1. No clause $C \in F$ is falsified by α.
2. The domain of α has size at most $|\mathrm{Dom}(\alpha)| \leq \ell$.
3. For every subassignment $\alpha' \subseteq \alpha$ it holds that $\alpha' \in \mathcal{D}$.
4. If $|\mathrm{Dom}(\alpha)| < \ell$, then for every variable x there exists an $\alpha' \in \mathcal{D}$ that assigns a value to x and extends α (i.e., $\alpha' \supseteq \alpha$).

In [AD08], Atserias and Dalmau proved the following tight connection between Duplicator winning strategies and resolution refutation width.

Theorem 15 ([AD08]). *The CNF formula F has a resolution refutation of width ℓ if and only if Duplicator has no winning strategy for the Boolean existential $(\ell+1)$-pebble game on F.*

The Duplicator strategy in Definition 14 has some similarities with the extendible family in Definition 5, which can be taken to suggest that there might be a relation between resolution width and PCR space. The main difference is that extendible families consist of sets of assignments in which we must be able to flip every variable, while Duplicator's strategy is built on fixed individual assignments. However, if we substitute every variable in F with a non-authoritarian function as defined in Definition 4, then it is straightforward to make the transition from fixed assignments to sets of flippable assignments.

Lemma 16. *Let F be a k-CNF formula and let f be a non-authoritarian function. If Duplicator wins the Boolean existential ℓ-pebble game on F, then there exists an $(\ell - k + 1)$-extendible family for $F[f]$.*

Proof. Let $\mathcal{D}$ be a winning Duplicator strategy for F. We will use $\mathcal{D}$ to construct an $(\ell - k + 1)$-extendible family $\mathcal{F}$ for the substituted formula $F[f]$. In what follows, let us denote by $\mathit{Vars}^d(x)$ the set of variables that we get when we substitute x by $f(x_1, \ldots, x_d)$ in F for some non-authoritarian function f of arity d.

For $x \in \mathit{Vars}(F)$, define $Q_x = \mathit{Vars}^d(x)$ and let $H_{x,\alpha} = \{\beta \mid \mathrm{Dom}(\beta) = Q_x \text{ and } f(\beta) = \alpha(x)\}$ be the set of all assignments over Q_x for which f evaluates to the value that α assigns to x. For any partial assignment $\alpha \in \mathcal{D}$ we let the corresponding structured assignment set $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha)$ be the pair consisting of $\mathcal{Q}_\alpha = \{Q_x \mid x \in \mathrm{Dom}(\alpha)\}$ and $\mathcal{H}_\alpha = \prod_{x \in \mathrm{Dom}(\alpha)} H_{x,\alpha}$. We define $\mathcal{F}$ to encompass all structured assignment sets $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha)$ corresponding to partial assignments $\alpha \in \mathcal{D}$ with $|\mathrm{Dom}(\alpha)| \leq \ell - k + 1$. We need to prove that $\mathcal{F}$ constructed in this way is an $(\ell - k + 1)$-extendible family with respect to $F' = \emptyset$.

By construction, for every $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha) \in \mathcal{F}$ we have that $\mathcal{Q}_\alpha$ is a partial partition and that the partial assignments $H_{x,\alpha} \in \mathcal{H}_\alpha$ assign to $Q_x \in \mathcal{Q}_\alpha$. Furthermore, $H_{x,\alpha}$ is flippable on Q_x. This is so since f is a non-authoritarian function, which means that for very variable in $x_i \in Q_x$ there exist assignments β_b, $b \in \{0, 1\}$, to Q_x such that $\beta_b(x_i) = b$ and $f(\beta_b) = \alpha(x)$. Hence, all $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha) \in \mathcal{F}$ are structured assignment sets.

The size condition $|\mathcal{Q}_\alpha| \leq \ell - k + 1$ in Definition 5 is clearly satisfied for all $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha) \in \mathcal{F}$, and respectfulness is vacuously true. To see that the restriction property also holds, consider any $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha) \in \mathcal{F}$ obtained from $\alpha \in \mathcal{D}$. For any subset $\mathcal{Q}' \subseteq \mathcal{Q}_\alpha$, let α' be the subassignment of α restricted to $\{x \mid Q_x \in \mathcal{Q}'\}$ and let $\mathcal{H}' = \prod_{Q_x \in \mathcal{Q}'} H_{x,\alpha} = \prod_{x \in \mathrm{Dom}(\alpha')} H_{x,\alpha'}$. Then since $\alpha' \in \mathcal{D}$ by Definition 14, it follows by the construction of $\mathcal{F}$ that $(\mathcal{Q}', \mathcal{H}{\restriction_{\mathcal{Q}'}}) = (\mathcal{Q}', \mathcal{H}') \in \mathcal{F}$ as required.

It remains to prove that $\mathcal{F}$ has the extension property. Let $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha) \in \mathcal{F}$ be such that $|\mathcal{Q}_\alpha| < \ell - k + 1$ and let C be a clause in $F[f]$. We need to argue that $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha)$ can be extended to satisfy C. Let $A \in F$ be the clause such that $C \in A[f]$, i.e., C is one of the clauses obtained when substituting f in A. If $\alpha \in \mathcal{D}$ satisfies A, it follows by construction that $\mathcal{H}_\alpha$ satisfies all of $A[f]$ and hence, in particular, C, and we are done. Otherwise, it follows from the definition of a winning Duplicator strategy and the fact that $|\alpha| \leq \ell - k$ that α can be extended to an assignment α' that queries

all of the (at most k) variables in A without falsifying the clause. Such an α' must satisfy A. Fix some variable $x^* \in \mathrm{Dom}(\alpha') \setminus \mathrm{Dom}(\alpha)$ such that α' satisfies A by assigning to x^*, and let α^* be the subassignment of α' with domain $\mathrm{Dom}(\alpha) \cup \{x^*\}$. This α' must be in $\mathcal{D}$ by Definition 14, and analogously to what was argued above it must hold that $\mathcal{H}_{\alpha^*}$ satisfies $C \in A[f]$. It is clear that $(\mathcal{Q}_\alpha, \mathcal{H}_\alpha) \preccurlyeq (\mathcal{Q}_{\alpha^*}, \mathcal{H}_{\alpha^*})$, and that $|\mathcal{Q}_{\alpha^*}| \leq |\mathcal{Q}_\alpha| + 1$. Hence, $\mathcal{F}$ satisfies extendibility, and the lemma follows. □

Combining Lemma 16 with the combinatorial characterization of width in Theorem 15 and the lower bound on space in terms of extendible families in Theorem 6, we obtain the first theorem claimed in Section 3.

Theorem 7 (restated). *Let F be a k-CNF formula and let f be any non-authoritarian function. Then*

$$Sp_{\mathcal{PCR}}(F[f] \vdash \bot) \geq \frac{W_{\mathcal{R}}(F \vdash \bot) - k + 1}{4} \ .$$

While it can be argued that this theorem might be interpreted as an indication that degree could be a lower bound for space in PCR, a more immediate and concrete consequence is that it gives us a way to prove the existence of formulas which have very small PCR refutations, but for which any refutation must have essentially maximal space. For polynomial calculus over fields of characteristic 2, we already have all the tools needed to argue this. In particular, the space lower bound needed follows immediately from Theorem 7 as described next.

Corollary 17. *Let G be an expander graph of bounded degree over n vertices, let χ be an odd-charge function on $V(G)$, and let G' be the multi-graph obtained by adding two copies of each edge in G. Then*

$$Sp_{\mathcal{PCR}}(\mathit{Ts}(G', \chi) \vdash \bot) = \Omega(n) \ .$$

Proof. As shown in [BW01], refuting Tseitin formulas over expander graphs requires linear width in resolution. It is not hard to see that substituting with XOR in a Tseitin formula over G is the same as considering the formula over the multi-graph with two copies of every edge. Thus $\mathit{Ts}(G', \chi)$ requires monomial space $\Omega(n)$ by Theorem 7, which is linear in the formula size if G is a constant-degree expander. □

As briefly discussed in Section 3, it is not hard to show that Tseitin formulas have small refutations in PCR (and even PC) over fields of characteristic 2, which yields Corollary 9 for this characteristic. However, this upper bound does not hold for characteristics distinct from 2. Therefore, we need to work with generalized version of Tseitin formulas and prove our results for such formulas instead. We do so in the next section.

5 Formulas With Small Proofs May Require Large Space

In Section 2 we defined Tseitin formulas as the CNF encoding of particular linear systems over $\mathbb{F}_2$. Here we consider a generalization over fields of any positive characteristic. Any such formula essentially defines an unsatisfiable linear system over $\mathbb{F}_p$ for some prime p. In order to efficiently encode this linear system as a CNF it is important that each equation mentions a small (for instance constant) number of variables: any equation over d variables can be encoded as a set of at most 2^d clauses with d literals each. In particular, Tseitin formulas are defined on directed graph as follows.

Definition 18. Let $G = (V, E)$ be a directed graph and $\chi : V \to \{0, 1, \ldots, p-1\}$ be a function. Identify every directed edge $(u, v) \in E$ with a variable $x_{(u,v)}$ and let $Mod^p_{v,\chi}$ denote the CNF encoding of the constraint that the number of *incoming* edges $x_{(u,v)}$ incident to a vertex $v \in V$ that are set to true, minus the number of *outgoing* edges $x_{(v,w)}$ set to true is equal to $\chi(v) \pmod p$. Then the *Tseitin formula* over G with respect to χ is $Ts^p(G, \chi) = \bigwedge_{v \in V} Mod^p_{v,\chi}$.

This formula is unsatisfiable when $\sum_v \chi(v) \not\equiv 0 \pmod p$. Compare Definition 2 with Definition 18: for $p = 2$ the definitions coincide because is such characteristic there is no difference between the contribution of the incoming and the outgoing edges. For $p = 2$ it is natural to define the formula in terms of undirected graphs, indeed. Not surprisingly, polynomial calculus over a field of characteristic p efficiently refutes unsatisfiable Tseitin formulas defined on sums modulo p.

Lemma 19. *Consider a directed graph $G = (V, E)$ with n vertices and constant degree, and a function $\chi : V \to \{0, 1, \ldots, p-1\}$ with $\sum_v \chi(v) \not\equiv 0 \pmod p$. The formula $Ts^p(G, \chi)$ has a tree-like polynomial calculus refutation of constant degree, size $\mathrm{O}(n \log n)$, and monomial space $\mathrm{O}(n)$.*

Furthermore, given any boolean function f on a constant number of variables, the result holds for the substituted formula $Ts^p(G, \chi)[f]$.

Proof. Let us first consider the case without substitution. Recall that true value is encoded as 0 and false as 1. In this encoding formula $Mod^p_{v,\chi}$ is equivalent to

$$\sum_{u\colon (u,v)\in E} (1 - x_{uv}) - \sum_{w\colon (v,w)\in E} (1 - x_{vw}) \equiv \chi(v) \pmod p \ . \tag{5.1}$$

The proof is based on the natural intuition that summing the equations (5.1) for all vertices in the graph results in a contradiction, since in the sum each variable appears twice: once with positive and once with negative sign. Fix an enumeration of $V = \{v_1, \ldots v_n\}$, and fix the following notation for partial sums:

$$S_{a,b} := \sum_{i=a}^{b} \left[\sum_{u:(u,v_i)\in E} (1 - x_{uv_i}) - \sum_{w:(v_i,w)\in E} (1 - x_{v_i w}) \right] \equiv \sum_{i=a}^{b} \chi(v_i) \pmod p \ . \tag{5.2}$$

We fix $t = 2^{\lceil \log n \rceil} < 2n$ and consider $S_{i,i}$ to be the equation "$0 = 0$" for all $n < i \leq t$. We set up a tree of height $\lceil \log n \rceil$, where leaves are labeled by equations $S_{i,i}$ and internal nodes are labeled by the sum of the two children labels (i.e., a node at level k is labeled by the equation $S_{i,i+2^k-1}$ for some i).

Each equation $S_{i,i}$ is derived from the encoding of $Mod^p_{v_i,\chi}$. This equation mention only a constant number of variables, so by implicational completeness of polynomial calculus (see Lemma 20) we have a derivation of constant space and size.

Equations in internal nodes are derived by summing the equations of the children. We derive all the equations on the tree in a bottom-up fashion. This concludes the refutation since the equation $S_{1,t}$ at the root is

$$\sum_{i=1}^{n} \left[\sum_{u:(u,v_i)\in E} (1 - x_{uv_i}) - \sum_{w:(v_i,w)\in E} (1 - x_{v_i w}) \right] \equiv \sum_{i=1}^{n} \chi(v_i) \pmod{p} \tag{5.3}$$

$$\sum_{(u,v)\in E} (1 - x_{uv}) - \sum_{(v,w)\in E} (1 - x_{vw}) \equiv \sum_{i=1}^{n} \chi(v_i) \pmod{p} \tag{5.4}$$

$$0 \equiv \sum_{i=1}^{n} \chi(v_i) \pmod{p} \tag{5.5}$$

Which is the end of the refutation, since $\sum_{i=1}^{n} \chi(v_i)$ is non-zero.

The size of the proof accounts $O(1)$ for the deduction of each $S_{i,i}$, and $O(n)$ for the total number of monomial at each level of the tree: at level k there are $\frac{t}{2^k}$ equations with at most $O(2^k)$ monomials. So the total size is as claimed.

Regarding the monomial space, notice that we need to keep simultaneously in memory only the equations of two adjacent levels, which have at most $O(n)$ monomials.

The degree of the refutation is $O(1)$ for the inference of each equation $S_{i,i}$. The rest of the proof has degree 1.

The case with substitution is similar: consider a substituting function f on a constant number of variables. There is a multilinear polynomial p_f which evaluates exactly as f on all $\{0,1\}$ inputs, and which mentions a constant number of monomials.

The substituted linear forms $S_{i,i}[f]$ are linear combinations of copies of p_f, so they have a constant number of variables each and their inference from $Mod^p_{v_i,\chi}[f]$ is doable in constant space, size and degree because of Lemma 20.

Once the equations $S_{i,i}[f]$ are derived, the refutation goes exactly as shown for the case with no substitution. From this point on the original refutation is linear; applying the trivial substitution to these proof lines increases the space, degree and size only by constant factors. □

For the sake of self-containment, we give a proof of the implicational completeness of polynomial calculus. This completes the proof of Lemma 19.

Lemma 20. *Consider a polynomial implication $p_1 \dots p_l \models p$ which is valid over $\{0,1\}$ assignments. Assume all involved polynomials collectively mention d variables and have degree $\mathrm{O}(d)$; then there is a PC proof of this implication in degree $\mathrm{O}(d)$, space $2^{\mathrm{O}(d)}$, and length $2^{\mathrm{O}(d)}$.*

Proof. Without loss of generality we assume that all polynomials are in multilinear form. So each of them has size at most 2^d and degree d. Let $\alpha = \{x_1 \mapsto v_1, \dots, x_d \mapsto v_d\}$ be an assignment; we define C_α as $\prod_i (v_i x_i + (1 - v_i)(1 - x_i))$, the polynomial which evaluates to 1 exactly on the assignment α. We list some useful observations:

Observation (1) is that given the axioms $\{x_i = v_i\}_{i \in [d]}$ and any polynomial q on variables $x_1, \dots, x_d$, it is possible to efficiently infer $q - \alpha(q) = 0$. We prove this by induction on the number of variables. If $d = 0$ then $q = \alpha(q)$. Now assume that $q - \alpha(q) = s + xt - \alpha(q)$. If we have deduced $q\restriction_{x=0} = s - \alpha(q)$ and we have the axiom x, we can easily infer xt and then $s + xt - \alpha(q)$. If we have deduced $q\restriction_{x=1}$ (which is $s + t - \alpha(q)$) and we have the axiom $x - 1$, we can easily infer $(x-1)t$ and then $s + t + (x-1)t - \alpha(q) = s + xt - \alpha(q)$. This derivation requires $\mathrm{O}(d)$ steps, one per variable, and both size and space are proportional to the number of monomials in q. The degree is equal to the degree of q plus d.

Observation (2) is that for any q on variables $x_1, \dots, x_d$, we can infer from Boolean axioms the polynomial $C_\alpha(q - \alpha(q))$, for every assignment α on such variables. The inference is in degree $\mathrm{O}(d)$, and length and space are $2^{\mathrm{O}(d)}$. It is immediate for the simple case $q = x_i$: each $C_\alpha(x_i - v_i)$ contains the factor $x_i^2 - x_i$ by construction. For any non-trivial q we apply the inference in Observation (1), with the caveat that each line is multiplied by C_α. The resulting polynomial is $C_\alpha(q - \alpha(q))$.

Observation (3) is that $\sum_{\alpha \in \{0,1\}^d} C_\alpha = 1$, and this is an easy induction over d (it also follows from the semantic of polynomials C_α).

We now see how to deduce $C_\alpha p$ for every assignment α. For α which satisfy p we derive $C_\alpha(p - 0)$ using observation (2). For α which falsify p, pick any falsified p_i and deduce both $C_\alpha(p_i - \alpha(p_i))$ and $C_\alpha p_i$, using observations (2) and multiplication rule, respectively. The sum is $C_\alpha \alpha(p_i)$, and since $\alpha(p_i)$ is a non-zero field element, we can multiply by $\frac{p}{\alpha(p_i)}$ to get $C_\alpha p$.

Having deduced all $C_\alpha p$ we can use observation (3) to infer p. Notice that we did 2^d inferences (one for each α), each of them of degree $\mathrm{O}(d)$ and each of them in space $2^{\mathrm{O}(d)}$. □

Now we have seen that (substituted) Tseitin formulas are easy to polynomial calculus under determined conditions. Nevertheless we can use the tools from Section 4 to show that even under such conditions, any refutation requires large space.

Theorem 21 (restatement of Theorem 8). *For $\mathbb{F}$ any field of characteristic p there is a family of k-CNF formulas F_n (where k depends on p) of size $\mathrm{O}(n)$ for which $Sp_{\mathcal{PCR}}(F_n \vdash \bot) = \Omega(n)$ over any field but which have tree-like PC refutations $\pi_n : F_n \vdash \bot$ over $\mathbb{F}$ of size $S(\pi_n) = \mathrm{O}(n \log n)$ and degree $Deg(\pi_n) = \mathrm{O}(1)$.*

Proof. The formula family we consider is based on Tseitin formulas over a family of Ramanujan graphs of constant degree. This is a family of simple graphs with good expansion properties; a construction is given in [Mor94]. Consider such a graph G on m vertices: set an arbitrary orientation on the edges, and consider any $\chi : [m] \to \{0, \ldots, p-1\}$ with $\sum_i \chi(i) \neq 0 \mod p$.

In Corollary 4.5 of [AR03], it is claimed that if G is a d-regular graph for d at least some constant value d_p, then $Ts^p(G, \chi)$ requires refutations of degree $\Omega(m)$ in polynomial calculus over any field of characteristic different from p.

Polynomial calculus simulates resolution over any characteristic, and the degree of the simulation is exactly the width of the simulated resolution proof. This implies that resolution requires width $\Omega(m)$ to refute the formula.

Fix $k = 2d$. We apply a XOR substitution on formula $Ts^p(G, \chi)$, and we get a k-CNF formula on $n = dm$ variables. Theorem 7 implies that any polynomial calculus (or PCR) refutation requires monomial space $\Omega(n)$, under any characteristic.

If the characteristic of the underlying field is p the upper bound follows by Lemma 19. □

6 PCR Space Lower Bounds for Tseitin Formulas

In the following exposition we assume that $G = (V, E)$ is a graph with connectivity expansion c and $\chi : V \to \{0, 1\}$ is a Boolean function. We call a pair (G, χ) a *charged graph*, and we say that a set of vertices U is even (odd) charged if $\sum_{v \in U} \chi(v)$ is even (odd). We denote the set of edges *incident* to a vertex v by $E(v)$ and extend the notation to sets of vertices. We write $\overline{\alpha}$ to denote the complementary assignment of α obtained by flipping the value of all variables in the domain $\mathrm{Dom}(\alpha)$.

Definition 22. The *charged graph induced by a partial assignment* α is $((V, E \setminus \mathrm{Dom}(\alpha)), \gamma)$, where $\gamma(v) = \chi(v) + \sum_{e \ni v}(1 - \alpha(e))$.

Observation 23. *The formulas $Ts((V, E \setminus \mathrm{Dom}(\alpha)), \gamma)$ and $Ts(G, \chi)\restriction_\alpha$ are equivalent. An assignment α satisfies the clauses $PARITY_{v,\gamma}$ if and only if the vertex v is isolated and even (as a singleton set) in the charged graph induced by α. In that case, we say that the assignment α satisfies the vertex v.*

Definition 24 (non-splitting assignment). A charged graph is *non-splitting* if all its connected components of size at most $n/2$ are even. A partial assignment α is *non-splitting* if the charged graph induced by α is non-splitting.

Observation 25. *The empty assignment is non-splitting for the charged graph (G, χ) if and only if (G, χ) is non-splitting. A connected graph is always non-splitting.*

Observation 26. *Suppose α is a partial assignment extending a partial assignment β (or conversely, $\beta = \alpha\restriction_D$ for some $D \subseteq \mathrm{Dom}(\alpha)$). If α is non-splitting, then so is β. In other words, "unsubstituting" an edge cannot result in an odd component that has size less than or equal to $n/2$ because component sizes can only increase.*

The key idea in the resolution space lower bound is that if a proof does not mention many edges, then it is possible to maintain a satisfiable assignment to the edges the proof mentions. This satisfiable assignment shifts the charged in the graph so that a contradiction only arises in vertices that the proof does not mention and leaves enough freedom to keep adding edges to the assignment unless the proof reaches a space threshold. Thus the proof is unable to derive a contradiction unless it mentions many edges at once.

The following lemma implements the charge shifting idea.

Lemma 27. *Let α be a non-splitting assignment. Let e be an edge. Let $D = \mathrm{Dom}(\alpha) \cup \{e\}$. If $|D| \leq c$ then we can extend α to some non-splitting assignment β such that $\mathrm{Dom}(\beta) = D$.*

Proof. Let (G', γ) be the charged graph induced by α. Let $e = (u, v)$. Let C be the connected component in G' that contains the vertices u and v. Let $\alpha_0 = \alpha \cup \{e \mapsto 0\}$ and $\alpha_1 = \alpha \cup \{e \mapsto 1\}$. Let (G'', γ_0) and (G'', γ_1) be the charged graph induced by α_0 and α_1 respectively. Observe that $\gamma_0(C) = \gamma_1(C) = \gamma(C)$.

If e is not a bridge, i.e., removing the edge e from G' does not disconnect C, then we can extend α to either α_0 or α_1. In this case there is no new component.

If e is a bridge, let C' and C'' be the components in G'' that e disconnects C into. If $\gamma(C)$ is even, either both $\gamma_0(C')$ and $\gamma_0(C'')$ are even, in which case we can extend α to α_1, or both $\gamma_0(C')$ and $\gamma_0(C'')$ are odd, in which case we can extend α to α_0 reversing both parities. In this case all new components are even.

Otherwise, since α is non-splitting, $|C| > n/2$. Since $|D| \leq c$, the graph G'' has a connected component larger than $n/2$. The graph G' cannot have two disjoint components both larger than $n/2$, so this large component is a subset of C; either C' or C''. Assume it is C' without loss of generality. Since C is odd, either $\gamma_0(C')$ is odd and $\gamma_0(C'')$ is even, in which case we can extend α to α_1, or $\gamma_0(C')$ is even and $\gamma_0(C'')$ is odd, in which case we can extend α to α_0 reversing both parities. In this case there is one new odd component, but it is larger than $n/2$. □

Corollary 28. *Let α be a non-splitting assignment. Let E be a set of edges. Let $D = \mathrm{Dom}(\alpha) \cup E$. If $|D| \leq c$ then we can extend α to some non-splitting assignment β such that $\mathrm{Dom}(\beta) = D$.*

To extend this idea to a PCR lower bound for space, and in particular to the framework of [BG13], we need to use assignments that are not only non-splitting but also resilient to flips of the values of some variables.

Observe that if all the edges along a cycle change their value, the graph induced by the cycle stays the same. The following definition will let us formalize this property. Recall the cartesian product notation for sets of assignments.

Definition 29 (Flipped assignments). Let α be a partial assignment and let $\mathcal{Q}$ be a (total) partition of $\mathrm{Dom}(\alpha)$. The set of *flipped assignments of α with respect*

to $\mathcal{Q}$ is the set of assignments given by

$$\mathit{Flip}(\mathcal{Q}, \alpha) = \prod_{Q \in \mathcal{Q}} \{\alpha\restriction_Q, \overline{\alpha}\restriction_Q\} \ .$$

Observation 30. *If* α *is an assignment over a cycle* C*, then* α *and* $\overline{\alpha}$ *induce the same charged graph. Therefore, if* $\mathcal{Q}$ *is a set of disjoint cycles, all the flipped assignments of some assignment* α *with respect to* $\mathcal{Q}$ *induce the same charged graph.*

Theorem 31 (Strengthening of Theorem 10). *Let* (G, χ) *be non-splitting charged graph of maximal degree* d *with connectivity expansion* c *such that a partition* M *of* E *into edge-disjoint cycles of length at most* b *exists. Then*

$$Sp_{\mathcal{PCR}}(\mathit{Ts}(G, \chi) \vdash \bot) \geq c/4b - d/8 \ .$$

Note that this is a strengthening of Theorem 10 since if G is connected then (G, χ) is trivially non-splitting for every χ.

Proof. By Theorem 6, it is sufficient to build an r-extendible family for $r = c/b - d/2$. Let $\mathcal{F}$ be the set of all pairs $(\mathcal{Q}, \mathcal{H}^\alpha)$ satisfying:

1. $\mathcal{Q} \subseteq M$ and $|\mathcal{Q}| \leq r$.
2. $\mathcal{H}^\alpha = \mathit{Flip}(\mathcal{Q}, \alpha)$, where α is any non-splitting assignment over $\bigcup \mathcal{Q}$.

Note that $\mathcal{Q}$ is a collection of edge-disjoint cycles and every $\mathcal{H}^\alpha$ consists of the some non-splitting assignment α and its flips over cycles. Each $(\mathcal{Q}, \mathcal{H}^\alpha) \in \mathcal{F}$ has many different representations, since $\mathcal{H}^\alpha = \mathcal{H}^\beta$ whenever $\beta \in \mathit{Flip}(\alpha, \mathcal{Q})$.

Let us show that $\mathcal{F}$ is an extendible family. First, pairs $(\mathcal{Q}, \mathcal{H}^\alpha)$ are $\mathcal{Q}$-structured by construction.

The empty assignment is non-splitting by Observation 25. So the family $\mathcal{F}$ is not empty because $(\emptyset, \mathcal{H}^\emptyset) \in \mathcal{F}$, where $\emptyset$ is the empty assignment.

Let us show that the family is closed under restriction. Consider any $(\mathcal{Q}, \mathcal{H}) \in \mathcal{F}$ and $\mathcal{Q}' \subseteq \mathcal{Q}$. Let $\alpha \in \mathcal{H}$, and let β be the restriction of α to $\bigcup \mathcal{Q}'$. By construction α is non-splitting, and restriction preserves the property of being non-splitting as noted in Observation 26, so $(\mathcal{Q}', \mathcal{H}^\beta) \in \mathcal{F}$. Finally $\mathcal{H}\restriction_{\mathcal{Q}'} = \mathit{Flip}(\mathcal{Q}, \alpha)\restriction_{\mathcal{Q}'} = \mathit{Flip}(\mathcal{Q}', \beta) = \mathcal{H}^\beta$.

Let us show that the family is closed under extension. Let $(\mathcal{Q}, \mathcal{H}) \in \mathcal{F}$ with $|\mathcal{Q}| < r$ and let $p \in \mathit{PARITY}_{v,\chi}$ for some vertex $v \in V$.

If $\mathcal{H}$ satisfies p we are done; otherwise we will extend a non-splitting assignment associated with $\mathcal{H}$.

Let $\alpha \in \mathcal{H}$ be a non-splitting assignment that does not satisfy p. Let $\mathcal{Q}_v = \{C \in M \mid v \in C\}$ be the cycles adjacent to v, and let $\mathcal{Q}_+ = \mathcal{Q}_v \setminus \mathcal{Q}$; we will see that $\mathcal{Q}_+$ is not empty, but we do not need to assume it now. Let $D = \mathrm{Dom}(\alpha) \cup \bigcup \mathcal{Q}_+$. By hypothesis $|\mathcal{Q} \cup \mathcal{Q}_+| < r + d/2$, and it follows that $|D| < c$. Thus we can apply Corollary 28 on α and $\bigcup \mathcal{Q}_+$ to extend α to a non-splitting assignment β over D.

The assignment β disconnects the component $\{v\}$ and is non-splitting, so it makes the component $\{v\}$ even. By Observation 23, β satisfies the vertex v. Note that β falsifies $p \cap \bigcup \mathcal{Q}$, the subclause of p with variables $\bigcup \mathcal{Q}$. If for all $C \in \mathcal{Q}_+$ the assignment β supersatisfies or falsified the subclause $p \cap C$, then there would be a non-splitting assignment in $\mathit{Flip}(\mathcal{Q}_+, \beta)$ that falsified p.

Let $C \in \mathcal{Q}_+$ be a cycle that contains one literal of p that β satisfies and one literal that β falsifies. Let $\mathcal{Q}' = \mathcal{Q} \cup \{C\}$ and let $\mathcal{H}' = \mathcal{H}^\beta$. By construction $(\mathcal{Q}', \mathcal{H}') \in \mathcal{F}$, and assignments in $\mathcal{H}'$ restricted to C satisfy p. □

Theorem 10 is somewhat restrictive, in that it requires us to partition *all* edges in the graph into short cycles. However, as the following corollary shows, it is enough to partition *most* of the edges.

Corollary 32. *Let (G, χ) be a non-splitting charged graph of maximal degree d with connectivity expansion c such that a partition M of E into edge-disjoint cycles of length at most b and an additional number of $t < c$ edges exist. Then*

$$Sp_{\mathcal{PCR}}(\mathit{Ts}(G, \chi) \vdash \bot) \geq (c - t)/4b - d/8$$

Proof. Let H be the graph obtained by removing the t extra edges. Note that the connectivity expansion of H is at least $c - t$. Corollary 28 on page 66 shows that there exists a non-splitting assignment α on $G \setminus H$. Observation 23 on page 65 implies that for some γ, (H, γ) is a non-splitting charged graph. By a restriction argument, any PCR refutation of a non-splitting Tseitin formula on G in space S can be translated to a PCR refutation of a non-splitting Tseitin formula on H in space at most S. Theorem 10 shows that $S \geq (c - t)/4b - d/8$. □

Application: Grid Graphs

There are families of graphs where we actually get matching upper and lower bounds for PCR space. One such family is square grids. For the following subsection let n be an even integer and denote $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$, the integers modulo n. The following defines a grid over a torus.

Definition 33 (Grid graph). The *grid graph* (or discrete torus) $T(n)$ is a 4-regular graph with vertices $V = \mathbb{Z}_n \times \mathbb{Z}_n$ and edges

$$E = \left\{ \big((i,j),(i+1,j)\big), \big((i,j),(i,j+1)\big) \,\middle|\, i,j \in V \right\} .$$

We order the vertices of $T(n)$ lexicographically: $(i,j) < (k,l)$ if $i < k$ or $i = k$ and $j < l$. The *predecessor* of a vertex $(i,j) \neq (1,1)$, denoted $\mathit{pred}(i,j)$, is the vertex immediately preceding (i,j) in this order.

We will explicitly refer to the edges we need to disconnect a set of vertices from a graph. This notion is known as edge boundary.

Definition 34. Let $G(V,E)$ be a graph and $U \subseteq V$ be a subset of vertices. The *edge boundary* of U is the set of edges $\partial_e(U) = \{(x,y) \in E : x \in U,\ y \notin U\}$.

We can find an upper bound on PC space by mentioning all the vertices in lexicographical order.

Lemma 35. *The space of refuting a Tseitin formula over the $n \times n$ grid graph for an odd charge function χ over characteristic 2 is $Sp_{\mathcal{PC}}(Ts(T(n),\chi) \vdash \bot) = \mathrm{O}(n)$.*

Proof sketch. Observe that for every set of vertices U it holds that $\sum_{e \in E(U)} e \equiv \sum_{e \in \partial_e(U)} e \pmod 2$, and that in PC over characteristic 2 this expression corresponds to the polynomial $\sum_{e \in \partial_e(U)} e$. Thus, we can express $\sum_{e \in E(U)} e \equiv \chi(U)$ in space $\partial_e(U)$. If we let $U_{ij} = \{(a,b) \in V \mid (a,b) \leq (i,j)\}$, the edge boundary of any U_{ij} is at most $2n+1$, so the monomial space of each of the polynomials $p_{ij} = \sum_{e \in \partial_e(U_{ij})} e - \chi(U_{ij})$ is at most $2n+1 = \mathrm{O}(n)$.

If we show how to derive the polynomials p_{ij} in lexicographical order in $\mathrm{O}(n)$ space, we will be done. And indeed, for any vertex (i,j) we can infer the polynomial $q_{ij} = \sum_{e \ni (i,j)} e - \chi(v)$ by downloading the 2^{d-1} axioms $PARITY_{(i,j),\chi}$ and adding all of them in constant space. To derive p_{ij} from $p_{pred(ij)}$ it is enough to add the polynomials $p_{pred(ij)}$ and q_{ij}. The maximum space is $Sp(p_{pred(ij)}) + Sp(p_{ij}) + \mathrm{O}(1) = \mathrm{O}(n)$. □

The connectivity expansion follows from the following isoperimetric inequality.

Theorem 36 ([BL91]). *Let U be a subset of vertices of $T(n)$ with $|U| \leq n^2/2$. Then*

$$|\partial_e(U)| \geq \min\{2n, 4|U|^{1/2}\}\ .$$

Corollary 37. *The connectivity expansion of $T(n)$ is $2n-1$.*

Proof. If we erase $2n-1$ or less edges from $T(n)$, then by Theorem 36 the largest region we can disconnect has size $|U| \leq \lfloor (2n-1)/4 \rfloor^2 < n^2/2$, so $c \geq 2n-1$. If we erase the $2n$ edges $\{((i,0),(i,1)) \mid i \in \mathbb{Z}_n\} \cup \{((i,n/2),(i,n/2+1)) \mid i \in \mathbb{Z}_n\}$ we obtain two connected components of size $n^2/2$, so $c < 2n$. □

The lower bound on PCR space follows.

Corollary 38. *The space of refuting a Tseitin formula over the $n \times n$ grid graph (over any characteristic) is $Sp_{\mathcal{PCR}}(Ts(T(n),\chi) \vdash \bot) = \Omega(n)$.*

Proof. Let us find a partition of the edges of $T(n)$. Let $C(i,j)$ be the set of edges of the cycle $\big((i,j),(i+1,j),(i+1,j+1),(i,j+1)\big)$. Then the set $M = \{C(i,j) \mid i+j \equiv 0 \pmod 2\}$ is a partition of the edges of $T(n)$ into edge-disjoint cycles of length 4. By Theorem 6, $Sp_{\mathcal{PCR}}(Ts(T(n),\chi) \vdash \bot) \geq (2n-9)/16$. □

Theorem 39. *The space of refuting a Tseitin formula over the $n \times n$ grid graph for an odd charge function χ over characteristic 2 is $Sp_{\mathcal{PCR}}(Ts(T(n),\chi) \vdash \bot) = \Theta(n)$.*

Application: Triangulations

Given a graph with good expansion, we can add a few edges to it and obtain a new graph whose Tseitin formula we can prove to be hard for PCR space. We already showed in Section 4 how to use a XOR substitution to obtain such a multi-graph; the following subsection shows how to obtain a simple graph. The proposed method is to convert every edge into a triangle, and a greedy strategy is enough as the following lemma shows.

Lemma 40. *Let G be a graph of order n, size m and maximal degree d. If T is an integer such that $T(n - 4d - 4(T + 1)) \geq m$ then there exists a simple graph H of maximal degree at most $2d + 2T$ which is a supergraph of G whose edges can be partitioned into disjoint triangles.*

Proof. Consider the algorithm that iteratively chooses any edge (x, y) not yet handled, chooses a vertex z not adjacent to any of the endpoints of minimal degree, and adds the two remaining edges (x, z) and (y, z) from the endpoints to the vertex.

We consider the new edges to be directed (from x and y to z) and the *indegree* and *outdegree* to refer to new edges only. The degree of a vertex is thus the sum of its initial degree, its indegree and its outdegree. Observe that at every step the outdegree of every vertex is at most its initial degree, which is at most d. When choosing the vertex z, we will choose the vertex of minimal *in*degree.

Assume that at some state S of the execution of the algorithm the maximal indegree is $2t$. We claim that the algorithm handles at least the next $n-4d-4(t+1)$ edges without the indegree exceeding $2(t + 1)$.

Indeed, consider the k-th edge (x, y) the algorithm visits after state S. Its endpoints are connected to at most $d + 2(t + 1) + d$ vertices each, which we discard as candidates for z, and at most $k - 1$ vertexes increased their indegree to $2(t + 1)$. There remain at least $n - 4d - 4(t + 1) - k + 1 \geq 1$ potential vertexes of indegree at most $2t$, and the greedy algorithm chooses one of these.

The initial indegree of all vertexes is 0. After handling all m edges, the maximal indegree increases at most T times, where T is such that

$$m \leq \sum_{t=0}^{T-1} n - 4d - 4(t + 1) = T(n - 4d - 4(T + 1)) \ . \tag{6.1}$$

□

In particular, if $d \leq n/4 - \sqrt{m} - 1$ such a T exists, and if $d = \mathrm{o}(n)$ the inequality (6.1) holds asymptotically for $T = \lceil \frac{d+1}{2} \rceil$. The lower bound on space follows by applying theorem Theorem 6 to the resulting supergraph and noting that the connectivity expansion cannot decrease.

Theorem 41. *Let G be a graph of maximal degree $d = \mathrm{o}(n)$ and connectivity expansion c. There exists a simple graph H of maximal degree at most $3d + 2$ which*

is a supergraph of G such that the space of refuting a Tseitin formula over H is at least $Sp_{\mathcal{PCR}}(Ts(H,\chi) \vdash \bot) \geq c/12 - (3d+2)/8$.

7 Cycle Partitions of Random Regular Graphs

Models of Random Regular Graphs

Let P_n be a sequence of probability spaces. A sequence of events E_n on P_n holds *asymptotically almost surely* if $\Pr[E_n] \longrightarrow 1$. In the sequel, we often abuse notation and say that an event is true asymptotically almost surely in a probability space, when we actually mean sequences of both. The probability space will depend on a parameter n.

Two probability spaces are *contiguous* if every event which holds asymptotically almost surely in one also holds asymptotically almost surely in the other; we will use the notation $A \approx B$ to denote that A and B are contiguous. Let $\mathcal{D}_d$ be the probability space of random d-regular graphs on n vertices, $\mathcal{H}+\mathcal{H}$ be the probability space of unions of (not necessarily disjoint) random Hamilton cycles on n vertices, and $\mathcal{H} \oplus \mathcal{H}$ be the probability space of unions of disjoint random Hamilton cycles on n vertices; $\mathcal{H} \oplus \mathcal{H}$ is obtained by conditioning $\mathcal{H}+\mathcal{H}$ upon the event that the two random Hamilton cycles are disjoint. Note that $\mathcal{H}+\mathcal{H}$ is a probability space on multi-graphs. Kim and Wormald [KW01] proved the following theorem (see also Wormald's survey [Wor99] and [JŁR00, §9.3–9.6]).

Theorem 42. *We have $\mathcal{D}_4 \approx \mathcal{H} \oplus \mathcal{H}$.*

We will need one more fact from [KW01], whose proof we only sketch.

Lemma 43. *If $G \sim \mathcal{H}+\mathcal{H}$ then $\Pr[G \text{ is simple}] \longrightarrow e^{-2}$.*

Proof sketch. Fix the first Hamilton cycle H_1. Let e_i be the (random) ith edge of the second Hamilton cycle H_2. It is easy to see that $\Pr[e_i \in H_1] = 2/(n-1)$, hence $\mathrm{E}[|H_1 \cap H_2|] \longrightarrow 2$. Moreover, one can show using Brun's sieve (for example [AS00, Theorem 8.3.1]) that the distribution of $|H_1 \cap H_2|$ is asymptotically Poisson; the required calculations are sketched in [KW01, §2(iii)]. Hence $\Pr[|H_1 \cap H_2| = 0] \longrightarrow e^{-2}$. □

Putting both facts together, we get the following result which will serve as our vantage point over random 4-regular graphs.

Lemma 44. *Suppose E is an event which holds asymptotically almost surely in $\mathcal{H}+\mathcal{H}$. Then E also holds asymptotically almost surely for random 4-regular graphs.*

Proof. Lemma 43 shows that E holds asymptotically almost surely in $\mathcal{H} \oplus \mathcal{H}$, and so in $\mathcal{D}_4$ by Theorem 42. □

Corollary 45. *A random 4-regular graph is connected asymptotically almost surely.*

Some Properties of Random Regular Graphs

For a graph $G = (V, E)$ and a subset U of the vertices, recall that $N(U)$ is the set of edges connecting U and $V \setminus U$. We say that the graph G is a δ-*expander* if for every set U of at most $|V|/2$ vertices, $|N(U)| \geq \delta|U|$. Note that our definition involves edge expansion. Bollobás [Bol88] proved the following fundamental result.

Theorem 46. *There is a constant c_1 such that asymptotically almost surely, a random 4-regular graph is a c_1-expander.*

In fact, we can choose any $c_1 < 2(1-\eta) \approx 0.4401$, where η is the unique positive solution of $(1-\eta)^{1-\eta}(1+\eta)^{1+\eta} = 2$. In particular, asymptotically almost surely a random 4-regular graph is a 0.44-expander.

The following lemma gives a lower bound on the connectivity expansion of a random 4-regular graph, defined in Definition 3.

Lemma 47. *There is a constant c_2 such that asymptotically almost surely, the connectivity expansion of a random 4-regular graph on n vertices is at least $c_2 n$.*

Proof. Let G be a random 4-regular graph. Theorem 46 shows that asymptotically almost surely, G is a c_1-expander. Suppose G has connectivity expansion s. There is a set W of s edges and an edge e such that $G \setminus W$ has a component of size larger than $n/2$, but $G \setminus (W \cup \{e\})$ has no component of size larger than $n/2$. Since e breaks the giant component into two components, $G \setminus (W \cup \{e\})$ must have a component U of size larger than $n/4$. Expansion shows that $|N(U)| \geq c_1|U| > (c_1/4)n$, and so $s = |W| \geq (c_1/4)n$. This shows that we can choose $c_2 = c_1/4$. □

Simple Lower Bound

In this section we prove that refuting a non-splitting Tseitin formula on a random 4-regular graph on n vertices requires space $\Omega\big(\sqrt{n/\log n}\big)$, asymptotically almost surely over the choice of the graph.

The idea is to prove that asymptotically almost surely, a random 4-regular graph on n vertices can be partitioned into cycles of length $\mathrm{O}\big(\sqrt{n \log n}\big)$. In order to prove that, it will be useful to consider a model related to $\mathcal{H} + \mathcal{H}$.

Let $[n] = \{1, \ldots, n\}$, and let S_n be the set of all permutations on $[n]$. Every permutation $\pi \in S_n$ determines a Hamilton cycle

$$H(\pi) = (\pi(1), \pi(2)), (\pi(2), \pi(3)), \ldots, (\pi(n-1), \pi(n)), (\pi(n), \pi(1)) \ . \tag{7.1}$$

(The cycle is undirected.) Let ι denote the identity permutation. We will consider the probability space $\mathcal{H}(\iota) + \mathcal{H}(\pi)$ formed by taking the union of $H(\iota)$ and $H(\pi)$, where π is chosen uniformly at random from S_n.

The idea of the proof is to divide $[n]$ into $\sqrt{n/\log n}$ blocks of length $\sqrt{n \log n}$. We will show that asymptotically almost surely, each block I_k contains a point t_k such that $s_k = \pi(t_k) \in I_k$. For any two adjacent blocks I_k, I_{k+1}, we can form a

cycle of length $\mathrm{O}\big(\sqrt{n \log n}\big)$ by pasting together the path from s_k to s_{k+1} in $H(\iota)$ and the path from $\pi(t_k)$ to $\pi(t_{k+1})$ in $H(\pi)$. As a result, the graph decomposes into $\sqrt{n/\log n}$ cycles of length $\mathrm{O}\big(\sqrt{n \log n}\big)$.

Let m be a parameter depending on n; in this section, we choose $m = \sqrt{n \log n}$, while in the next section, we choose $m = C\sqrt{n}$. For simplicity, we assume that m and n/m are both integers. We partition $[n]$ into n/m blocks $I_1, \ldots, I_{n/m}$ of size m: $I_k = \{(k-1)m + 1, \ldots, (k-1)m + m\}$. Let B_k be the event that $\pi(I_k) \cap I_k = \emptyset$. We think of B_k as a bad event, and our goal in this section is to show that asymptotically almost surely, none of the B_k happen. In order to show this, we estimate the probability that B_k happens.

Lemma 48. *For $k \in [n/m]$, $\Pr[B_k] \leq e^{-m^2/n}$.*

Proof. Using $1 - x \leq e^{-x}$, we calculate

$$\Pr[B_k] = \prod_{i=0}^{m-1} \left(1 - \frac{m}{n-i}\right) \leq \left(1 - \frac{m}{n}\right)^m \leq e^{-m^2/n} \ . \tag{7.2}$$

□

If $\overline{B_k}$ holds, we define t_k to be the first point in I_k such that $\pi(t_k) \in I_k$, and let $s_k = \pi(t_k)$.

Lemma 49. *Suppose $\overline{B_k}$ and $\overline{B_{k+1}}$ both hold (indices taken modulo n/m). Define a cycle C_k by taking two paths P_k^ι, P_k^π from $s_k = \pi(t_k)$ to $s_{k+1} = \pi(t_{k+1})$, one from each of the two Hamilton cycles:*

$$P_k^\iota = (s_k, s_k + 1), (s_k + 1, s_k + 2), \ldots, (s_{k+1} - 1, s_{k+1}) \ ,$$
$$P_k^\pi = (\pi(t_k), \pi(t_k + 1)), (\pi(t_k + 1), \pi(t_k + 2)), \ldots, (\pi(t_{k+1} - 1), \pi(t_{k+1})) \ .$$

The length of C_k is at most $4m$.

Proof. Assume for simplicity that $k \neq n/m$. Then $s_k, t_k \geq (k-1)m + 1$ and $s_{k+1}, t_{k+1} \leq km + m$. The length of C_k is $(s_{k+1} - s_k) + (t_{k+1} - t_k) \leq 4m - 2$. □

If none of the bad events happen, then the cycles $C_1, \ldots, C_{n/m}$ cover all of the graph. Choosing m accordingly, we can ensure that this happens asymptotically almost surely.

Lemma 50. *Let $m = \sqrt{n \log n}$. Asymptotically almost surely, a graph chosen according to $\mathcal{H}(\iota) + \mathcal{H}(\pi)$ decomposes into n/m cycles of size at most $4m$.*

Proof. According to Lemma 48, for each $k \in [n/m]$, $\Pr[B_k] \leq e^{-\log n} = 1/n$. A union bound shows that asymptotically almost surely, none of the B_k happen. Lemma 49 shows that the graph decomposes into n/m cycles of size at most $4m$. □

The lemma easily implies the lower bound.

Theorem 51. *Asymptotically almost surely, the space required to refute in PCR any Tseitin formula on a random 4-regular graph on n vertices is $\Omega\big(\sqrt{n/\log n}\big)$.*

Proof. For reasons of symmetry, Lemma 50 implies that asymptotically almost surely, a graph chosen according to $\mathcal{H}+\mathcal{H}$ decomposes into cycles of size at most $4\sqrt{n\log n}$. Lemma 47 shows that asymptotically almost surely, the connectivity expansion of the graph is at least $\Omega(n)$. Corollary 45 shows that asymptotically almost surely, the graph is connected, and so the Tseitin formula is non-splitting. Hence Theorem 10 gives a lower bound of $\Omega(\sqrt{n/\log n})$. □

Improved Lower Bound

In this section we improve the results of Section 7 by showing that refuting a non-splitting Tseitin formula on a random 4-regular graph on n vertices requires space $\Omega\big(\sqrt{n}\big)$, asymptotically almost surely over the choice of the graph.

We use the general method of Section 7, with a different choice of m, namely $m = C\sqrt{n}$ for some constant C to be determined later. Thinking of B_k as an indicator variable, let $B = \sum_{k=1}^{n/m} B_k$. Lemma 48 shows that $\mathrm{E}[B] \leq e^{-C^2}(n/m)$. We will show that asymptotically almost surely, $B \leq 2e^{-C^2}(n/m)$. This implies that the cycles C_k together cover most of the graph, and therefore Corollary 32 applies. The difficult part of the proof is showing that B is concentrated around its mean.

Let $p = \Pr[B_k]$ (all the probabilities are the same). We need the following strengthening of Lemma 48.

Lemma 52. *Let $p = \Pr[B_k]$, where B_k is the event that $I_k \cap \pi(I_k) = \emptyset$. As $n \longrightarrow \infty$, we have that $p \longrightarrow e^{-C^2}$.*

In order to show that B is concentrated around its mean, we show that for $k \neq l$, the events B_k and B_l are asymptotically negatively correlated.

Lemma 53. *For every $k \neq l \in [n/m]$, $\Pr[B_k \wedge B_l] \leq p^2 + \mathrm{o}(1)$.*

We prove both lemmas below, but first, let us see how they imply the desired result. The idea is that since any two bad events are asymptotically negatively correlated, the variance of B is small, and so Chebyshev's inequality shows that B is concentrated around its mean.

Lemma 54. *Asymptotically almost surely, $B \leq 2e^{-C^2}(n/m)$.*

Proof. We have $\mathrm{E}[B] = (n/m)p$ and

$$\begin{aligned}\mathrm{Var}(B) &= \mathrm{E}[B^2] - (\mathrm{E}[B])^2 \\ &= (n/m)p + (n/m)(n/m-1)(p^2+\mathrm{o}(1)) - (n/m)^2p^2 \\ &= (n/m)p(1-p) + \mathrm{o}\big((n/m)^2\big)\ ,\end{aligned}$$

using Lemma 53. Chebyshev's inequality shows that

$$\Pr[|B - \mathrm{E}[B]| > \mathrm{E}[B]] \leq \frac{\mathrm{Var}(B)}{\mathrm{E}[B]^2} \leq \frac{(n/m)p + \mathrm{o}\big((n/m)^2\big)}{(n/m)^2 p^2} = \mathrm{o}(1) \ , \tag{7.3}$$

since $p = \Omega(1)$ by Lemma 52. Therefore asymptotically almost surely, $B \leq 2\,\mathrm{E}[B] = 2(n/m)p \leq 2e^{-C^2}(n/m)$, using Lemma 48. □

The preceding lemma shows that the fraction of bad indices (indices k such that B_k holds) is small. Say that a block I_k is *good* if $\overline{B_k}$ and $\overline{B_{k+1}}$ both hold, and say that it is *supergood* if both I_{k-1} and I_k are good. Lemma 49 associates a cycle C_k with each good block I_k. If I_k is supergood, then the cycles C_{k-1} and C_k together cover the entire stretch of I_k, as the following lemma shows.

Lemma 55. *Suppose that block I_k is supergood. Then the union of the cycles C_{k-1}, C_k given by Lemma 49 contains the path of length m from $\min I_k$ to $\min I_{k+1}$ in $H(\iota)$, as well as the path of length m from $\pi(\min I_k)$ to $\pi(\min I_{k+1})$ in $H(\pi)$.*

Proof. The cycle C_{k-1} contains the path from $s_{k-1} < \min I_k$ to s_k in $H(\iota)$. The cycle C_k contains the path from s_k to $s_{k+1} \geq \min I_{k+1}$ in $H(\iota)$. Both paths together cover the path from $\min I_k$ to $\min I_{k+1}$ in $H(\iota)$. The argument for $H(\pi)$ is identical. □

We can now prove an analogue of Lemma 50.

Lemma 56. *Let $m = C\sqrt{n}$. Asymptotically almost surely, a graph chosen according to $\mathcal{H}(\iota) + \mathcal{H}(\pi)$ decomposes into cycles of size at most $4m$ and t additional edges, where $t \leq 12e^{-C^2}n$.*

Proof. Lemma 54 shows that asymptotically almost surely, all but $6e^{-C^2}(n/m)$ of the n/m blocks $I_1, \ldots, I_{n/m}$ are supergood. Let $\mathcal{C}$ be the (disjoint) union of all cycles C_k constructed using Lemma 49 for all good blocks I_k. The lemma shows that each cycle has size at most $4m$. Lemma 55 shows that $\mathcal{C}$ contains all but at most $12e^{-C^2}n$ edges of the graph. □

Replacing Theorem 10 with its corollary, Lemma 56 easily implies the lower bound.

Theorem 57. *Asymptotically almost surely, the space required to refute in PCR any Tseitin formula on a random 4-regular graph on n vertices is $\Omega\big(\sqrt{n}\big)$.*

Proof. For reasons of symmetry, Lemma 56 implies that asymptotically almost surely, a graph chosen according to $\mathcal{H} + \mathcal{H}$ decomposes into cycles of size at most $4C\sqrt{n}$ and t additional edges, where $t \leq 12e^{-C^2}n$. For an appropriate choice of C, $t \leq (c_2/2)n$. Lemma 47 shows that asymptotically almost surely, the connectivity expansion of the graph is at least $c_2 n$. Corollary 45 shows that asymptotically almost surely, the graph is connected, and so the Tseitin formula is non-splitting. Hence Corollary 32 gives a lower bound of $\Omega\big(\sqrt{n}\big)$. □

Technical Lemmas

We now turn to the proofs of Lemma 52 and Lemma 53. We start with the former.

Proof of Lemma 52. It is easy to check that for $0 \le x \le 1/2$, $1 - x \ge e^{-x-x^2}$. Therefore for large enough n,

$$p = \prod_{i=0}^{m-1} \left(1 - \frac{m}{n-i}\right) \ge \left(1 - \frac{m}{n-m}\right)^m \ge \exp\left[-\frac{m^2}{n-m} - \frac{m^3}{(n-m)^2}\right] \quad . \tag{7.4}$$

For large enough n, $m \le n/2$, and so $m^2/(n-m) = m^2/n + m^3/(n(n-m)) \le m^2/n + 2m^3/n^2$. Similarly, $m^3/(n-m)^2 \le 4m^3/n^2$. Therefore, using $e^{-x} \ge 1 - x$,

$$p \ge \exp\left[-\frac{m^2}{n} - 6\frac{m^3}{n^2}\right] = \exp\left[-C^2 - \frac{6C^3}{\sqrt{n}}\right] \ge e^{-C^2}\left(1 - \frac{6C^3}{\sqrt{n}}\right) \quad . \tag{7.5}$$

Hence $\liminf p \ge e^{-C^2}$. Lemma 48 shows that also $\limsup p \le e^{-C^2}$. □

The proof of Lemma 53 is more involved. Recall that the lemma claims that the events B_k and B_l are asymptotically negatively correlated. In fact, they are asymptotically uncorrelated. Recall that $\Pr[B_k]$ is roughly equal to e^{-C^2}. Given the value of π on I_k, the probability $\Pr[B_l]$ depends on $|\pi(I_k) \cap I_l|$. Typically, this intersection will be very small, and so $\Pr[B_l]$ is also roughly equal to e^{-C^2}.

We will show that $|\pi(I_k) \cap I_l|$ is typically small using an extension of the well-known Chernoff bound due to Kabanets and Impagliazzo [IK10, Theorem 1.1], attributed there to Panconesi and Srinivasan [PS97].

Theorem 58. *Let $X_1, \ldots, X_r$ be Boolean random variables such that for any set $S \subseteq [r]$, $\Pr[\bigwedge_{i \in S} X_i] \le \delta^{|S|}$. Then for $\gamma \ge \delta$,*

$$\Pr\left[\sum_{i=1}^{r} X_i \ge \gamma r\right] \le e^{-2r(\gamma-\delta)^2} \quad .$$

The following lemma applies this bound to our situation (in an abstracted version).

Lemma 59. *Let a, b, c be integers such that $a \ge b, c$, and let T be a random subset of $[a]$ of size b. For all $\rho \ge 1$,*

$$\Pr[|T \cap [c]| \ge \rho(bc/a)] \le e^{-2c(\rho-1)^2(b/a)^2} \quad .$$

Proof. For $i \in [c]$, let X_i be the event that $i \in T$. For $S \subseteq [c]$ such that $|S| \le b$,

$$\Pr_T[S \subseteq T] = \frac{\binom{a-|S|}{b-|S|}}{\binom{a}{b}} = \prod_{k=0}^{|S|-1} \frac{b-k}{a-k} \le \left(\frac{b}{a}\right)^{|S|} \quad . \tag{7.6}$$

Therefore we can apply Theorem 58 with $r = c$, $\delta = b/a$ and $\gamma = \rho(b/a)$. □

We can now prove Lemma 53.

Proof of Lemma 53. We will show that $\Pr[B_l \mid B_k] \leq p + \mathrm{o}(1)$. This implies that $\Pr[B_k \wedge B_l] = \Pr[B_k]\Pr[B_l \mid B_k] \leq p(p + \mathrm{o}(1)) = p^2 + \mathrm{o}(1)$.

Assuming the event B_k happens, $\pi(I_k)$ is a random subset of $[n] \setminus I_k$ of size m. Plugging $a = n - m$ and $b = c = m$ in Lemma 59, we deduce that for all $\rho \geq 1$,

$$\Pr[|\pi(I_k) \cap I_l| \geq \rho C^2 \mid B_k] \leq e^{-2(\rho-1)^2 m(m/(n-m))^2} \tag{7.7}$$

$$\leq e^{-2(\rho-1)^2 m^3/n^2} = e^{-2C^3(\rho-1)^2/\sqrt{n}} \ . \tag{7.8}$$

Hence with probability $1 - \mathrm{o}(1)$ given B_k, $D \triangleq |\pi(I_k) \cap I_l| \leq \sqrt{m \log m}$. Now

$$\Pr[B_l \mid D = d] = \prod_{i=0}^{m-1} \left(1 - \frac{m-d}{n-i}\right) \leq \left(1 - \frac{m-d}{n}\right)^m \leq e^{-m(m-d)/n} \ . \tag{7.9}$$

For $0 \leq x \leq 1$, one can check that $e^x \leq 1 + 2x$. Hence

$$\Pr[B_l \mid D \leq \sqrt{m \log m}] \leq e^{-m(m-\sqrt{m \log m})/n} \tag{7.10}$$

$$= e^{-C^2 + m\sqrt{m \log m}/n} \leq e^{-C^2}\left(1 + \frac{2m\sqrt{m \log m}}{n}\right) \ . \tag{7.11}$$

Using Lemma 52, we deduce that $\Pr[B_l \mid D \leq \sqrt{m \log m}] \leq e^{-C^2} + \mathrm{o}(1) = p + \mathrm{o}(1)$. We conclude that $\Pr[B_l \mid B_k] = p + \mathrm{o}(1)$ and so $\Pr[B_k \wedge B_l] = p^2 + \mathrm{o}(1)$. □

Regular Graphs of Degree Larger Than Four

Wormald [Wor99, Corollary 4.17] showed that when $d > 4$, a random d-regular graph can be obtained (up to contiguity) by taking the disjoint union of a random 4-regular graph and a random $(d-4)$-regular graph, a result summarized in the following theorem (see also [JŁR00, Corollary 9.44]).

Theorem 60. *For $d > 4$ we have $\mathcal{D}_d \approx \mathcal{D}_4 \oplus \mathcal{D}_{d-4}$. Furthermore, the probability that a uniformly random 4-regular graph and a uniformly random $(d-4)$-regular graph do not intersect tends to a positive constant.*

A Tseitin formula on a random d-regular graph generated according to $\mathcal{D}_4 \oplus \mathcal{D}_{d-4}$ is harder to refute than a Tseitin formula on a random 4-regular graph, and so we can generalize Theorem 57 to random d-regular graphs for arbitrary $d \geq 4$.

Theorem 61 (restatement of Theorem 11). *Let $d \geq 4$. Asymptotically almost surely, the space required to refute in PCR any Tseitin formula on a random d-regular graph on n vertices is $\Omega(\sqrt{n})$.*

Proof. If $d = 4$ then Theorem 57 already applies, so assume $d > 4$. Let G_1 be a random 4-regular graph, and let G_2 be a random $(d-4)$-regular graph. The graph $G = G_1 + G_2$ is distributed according to $\mathcal{D}_4 + \mathcal{D}_{d-4}$. We show below that asymptotically almost surely, the space required to refute in PCR any Tseitin formula on G is $\Omega(\sqrt{n})$. Since G_1 and G_2 are disjoint with constant probability according to Theorem 60, the theorem follows.

Let α be an arbitrary assignment to the edges of G_2. Observation 23 on page 65 shows that for every function f, $Ts(G,\chi)\restriction_\alpha = Ts(G_1,\gamma)$ for some other function γ. By a restriction argument, any PCR refutation of $Ts(G,\chi)$ in space S can be translated to a PCR refutation of $Ts(G_1,\gamma)$ in space at most S. Theorem 57 on page 75 shows that asymptotically almost surely, we must have $S = \Omega(\sqrt{n})$. □

8 Current Techniques and the Functional Pigeonhole Principle

We now discuss the intrinsic limitations of the techniques employed so far. In Section 8 we show that Bonacina-Galesi framework does not allow to prove PCR space lower bounds for an interesting formula like functional pigeonhole principle. In Section 8 we show that restricting to PC does not make the problem easier.

FPHP Formulas Do Not Have Extendible Families

One of the limits of the Bonacina-Galesi framework is that we cannot apply it to formulas for which fixing a small set of variables causes a lot of unit clauses propagation. Indeed, most of the lower bound strategies in this paper aim to control this phenomenon (see for example Lemma 16). For the functional pigeonhole principle these strategies do not work, as we now prove.

Definition 62. The *functional pigeonhole principle* on m pigeons and n holes is the formula defined on variables x_{ij} for $i \in [m]$ and $j \in [n]$, made of the following clauses:

$$\bigvee_{j\in[n]} x_{ij} \qquad \text{for all } i \in [m]; \qquad \text{(pigeon axioms)}$$

$$\neg x_{ij} \vee \neg x_{i'j} \qquad \text{for any } i \neq i' \in [m] \text{ and } j \in [n]; \qquad \text{(hole axioms)}$$

$$\neg x_{ij} \vee \neg x_{ij'} \qquad \text{for any } i \in [m] \text{ and } j \neq j' \in [n]. \qquad \text{(functional axioms)}$$

It is already known that this formula requires large space in resolution [BW01, AD08]. It is natural to suspect that this formula is hard in terms of monomial space as well. However, the Bonacina-Galesi framework is not strong enough to prove it.

Theorem 63 (restatement of Theorem 12). *There is no r-extendible family for $FPHP^m_n$ for $r > 1$.*

Proof. Assume that there is an r-extendible family $\mathcal{F}$ for the formula $FPHP_n^m$ which respects some satisfiable $F' \subseteq FPHP_n^m$, for $r > 1$.

Let C be any clause in $FPHP_n^m \setminus F'$; such clause exists because $FPHP_n^m$ is a contradiction. The extension property of $\mathcal{F}$ implies that there is a pair $(\{Q_1\}, H_1) \in \mathcal{F}$, where H_1 satisfies C.

Recall that 0 encodes true, and 1 encodes false. Pick a variable x_{ij} in Q_1. In H_1 there is at least one partial assignment for which $x_{ij} = 0$, and for any such assignment it holds that $x_{i'j} = 1$ and $x_{ij'} = 1$ for all $i' \neq i$ and $j' \neq j$, otherwise an initial clause would be false.

Indeed, fix v to be any of these variables (either $x_{i'j}$ or $x_{ij'}$); the clause $\neg x_{ij} \vee \neg v$ is an axiom. If $v \notin Q_1$ then this clause is not in F' because of the respectfulness of $\mathcal{F}$, and furthermore there is at least one assignment in H_1 which does not satisfy it (i.e., any assignment with $x_{ij} = 0$). The extension property of $\mathcal{F}$ guarantees that there is $(\{Q_1, Q_2\}, H_1 \times H_2) \in \mathcal{F}$ with $v \in Q_2$, such that $H_1 \times H_2$ satisfies $\neg x_{ij} \vee \neg v$. But this contradicts the fact that $H_1 \times H_2$ contains the assignment $\{x_{ij} = 1, v = 1\}$, which falsifies $\neg x_{ij} \vee \neg v$.

It follows that $\{x_{i'j}, x_{ij'} \mid i' \neq i \text{ and } j' \neq j\} \subseteq Q_1$, and that H_1 satisfies all axioms involving either pigeon i or hole j. We have just shown that assuming some $x_{ij} \in Q_1$, we get $\{x_{i'j}, x_{ij'} \mid i' \in [m], j' \in [n]\} \subseteq Q_1$. This choice was arbitrary, so it follows that for any $i \in [m], j \in [n]$, the variable x_{ij} is in Q_1. In other words, Q_1 contains all the variables. Since $FPHP_n^m \setminus F'$ is contradictory, every assignment in H_1 falsifies some clause, and so the extension property fails for any such clause. We conclude that $FPHP_n^m$ has no 2-extendible family. □

Formulas with Equal PC and PCR Space Complexities

Although finding an r-extendible family for the functional pigeonhole principle (and hence proving a space lower bound) is not feasible, we might try and prove a weaker PC space lower bound. However, as we have pointed out in Section 3, in the case of functional pigeonhole principle this makes no difference. In this section, we prove formally this result for a broader class of formulas that is captured by the following definition.

Definition 64. We say that a CNF formula F is *totally weight constrained* if for every variable x appearing in F there exists a clause $C_x \in F$ with the following properties:

1. All literals in C_x are positive;
2. x is one of the variables appearing in C_x;
3. For every two distinct variables y, z appearing in C_x, clause $\overline{y} \vee \overline{z}$ is in F.

For each variable x we refer to C_x as the *x-neighborhood clause*.

In such formulas each negative literal can be replaced with a clause/monomial consisting of only positive literals that has the same semantic meaning. Thus, we can turn a PCR refutation into a PC refutation without any substantial loss of space. In order for us to be able to show that such a refutation is a valid PC refutation we need to show that there are PC derivations of these monomials that use small space.

Theorem 65. *For a totally weight constrained CNF formula F, where each clause has a costant number of negative literals, it holds that $Sp_{PC}(F \vdash \bot) = \Theta(Sp_{PCR}(F \vdash \bot))$.*

Proof. We can easily see that PCR simulates PC with only a constant loss in space. The only problem in the simulation could arise when downloading an axiom that has negative literals. Nevertheless, it is not hard to prove that PCR can expand every axiom to its PC form while respecting the stated space bound.

In the other direction, we prove that PC can simulate a PCR refutation of F. Let π be a PCR refutation of F in space at most s. As F is a totally weight constrained formula, for every variable x we can fix its x-neighborhood clause C_x. Let us denote by $N(x)$ the set of variables from C_x excluding x. We transform the PCR refutation π into a PC refutation by replacing each negative literal $\overline{x}$ with the monomial $\prod_{y \in N(x)} y$. Obviously this transformation preserves space and we need to show that the transformed configurations form a backbone of a valid PC refutation.

If the PCR refutation deletes a polynomial, we delete the appropriate transformed polynomial from the configuration in the PC refutation. Similarly, in the case of linear combination steps we just deduce the linear combination of the transformed polynomials. Hence, these two types of steps can be done without any loss in space. In the case of multiplication with a literal, if the literal is positive we multiply the appropriate transformed polynomial with the same literal. Otherwise, the literal is negative and we multiply the polynomial with all the variables in $N(x)$, where $\overline{x}$ is the literal, while making sure to delete the intermediate polynomials when they are no longer needed. In this way we derive the transformed polynomial in at most $O(s)$ space.

The axiom download steps are the only ones that remain. In the case of Boolean axiom download, if we downloaded an axiom for a positive literal, we just download the appropriate axiom in the PC refutation. Otherwise, the Boolean axiom corresponds to some negative literal $\overline{x}$ and we need to derive the polynomial $\prod_{y \in N(x)} y^2 - \prod_{y \in N(x)} y$. This is done by downloading the Boolean axioms for each $y \in N(x)$ and combining them to get the transformed polynomial. Let $B^2 - B$ be one of the intermediate polynomials in the derivation of the transformed Boolean axiom, where B is a monomial formed by multiplying the variables in some subset of $N(x)$. Then, for some variable y not mentioned in B, we derive $(By)^2 - By$ by downloading $y^2 - y$ and taking the linear combination of $y(B^2 - B)$ and $B^2(y^2 - y)$. This PC derivation uses $O(1)$ more monomials than the PCR axiom download.

When the PCR proof downloads the complementarity axiom $1 - x - \overline{x}$, the corresponding PC proof needs to derive the polynomial $1 - x - \prod_{y \in N(x)} y$. Let

$N(x) = \{y_1, \ldots, y_l\}$. We derive the transformed polynomial by successively deriving polynomials

$$T(i) = \prod_{k=i+1}^{l} y_k - x \prod_{k=i+1}^{l} y_k - \prod_{k} y_k \ , \tag{8.1}$$

for $i = 1, \ldots, l$. Note that $T(l)$ is our transformed polynomial. The first $T(1)$ in the PC proof can be derived by downloading the axiom $(1-x)(1-y_1)$ and multiplying it with variables $y_2, \ldots, y_l$ in order to get $T(1) + x\prod_k y_k$. Subtracting from it the x-neighborhood clause $C_x = x\prod_k y_k$ we get $T(1)$.

We proceed to derive $T(i+1)$ from $T(i)$ for all i. Similarly as before, we start by downloading the axiom $(1-x)(1-y_{i+1})$ and multiplying it with variables $y_{i+2}, \ldots, y_l$ in order to get $T(i+1)-T(i)$. Adding this polynomial to $T(i)$ we derive the $(i+1)$st polynomial $T(i+1)$ in our derivation of the transformed complementarity axiom. This PC derivation uses O(1) more monomials than the PCR proof and all axioms of the form $(1-x)(1-y_i)$ exist because F is totally weight constrained.

In the case of axiom download step for a clause axiom, we again have two cases. If all literals of the axiom are positive we download the corresponding axiom in the PC proof. Otherwise, we can write the axiom as $\overline{x_1} \cdots \overline{x_s} \cdot x_{s+1} \cdots x_l$, where s is the number of its negative literals. Let us denote by $A(i)$ the polynomial

$$A(i) = \prod_{y_1 \in N(x_1)} y_1 \cdots \prod_{y_i \in N(x_i)} y_i(1-x_{i+1}) \cdots (1-x_s)x_{s+1} \cdots x_l \ , \tag{8.2}$$

where i ranges over $0, \ldots, s$. Note that $A(0)$ is the original PC axiom, while $A(s)$ is the transformed axiom that we want to derive. Also, let us denote by $R(i)$ the polynomial

$$R(i) = \prod_{y_1 \in N(x_1)} y_1 \cdots \prod_{y_{i-1} \in N(x_{i-1})} y_{i-1} \cdot (1-x_{i+1}) \cdots (1-x_s)x_{s+1} \cdots x_l \ , \tag{8.3}$$

for i ranging from 1 to s, that is $A(i) = R(i)\prod_{y_i \in N(x_i)} y_i = R(i+1)(1-x_{i+1})$.

We first derive $A(1)$ by deriving the transformed complementarity axiom $1 - x_1 - \prod_{y_1 \in N(x_1)} y_1$ for the variable x_1 and multiplying it with $R(1)$ in order to get $A(0) - A(1)$. Now we can get $A(1)$ by subtracting the derived polynomial from the PC axiom $A(0)$.

We proceed to derive $A(s)$ by deriving $A(i+1)$ from $A(i)$ for all i from 1 to $s-1$. This is again done by first deriving the appropriate complementarity axiom $1 - x_{i+1} - \prod_{y_{i+1} \in N(x_{i+1})} y_{i+1}$ and multiplying it by $R(i+1)$ in order to get $A(i) - A(i+1)$. Subtracting the derived polynomial from previously derived $A(i)$, we get the $(i+1)$st polynomial in our derivation. These steps use $O(2^s)$ monomials, which is constant by the theorem hypothesis, and the PC derivation of the transformed axiom uses at most O(1) monomials more than the PCR axiom download step.

Hence, the theorem follows. Also, although we have ignored the constants involved in the simulation, these constants can be computed explicitly and are small.

The only possible exception is the additive constant $O(2^{s^*})$, where s^* is the largest number of negative literals in a clause of F. □

An obvious example of the totally weight constrained formula is the functional pigeonhole principle.

Corollary 66 (Restatement of Theorem 13). *It holds that*

$$Sp_{PCR}(FPHP_n^m \vdash \bot) = \Theta(Sp_{PC}(FPHP_n^m \vdash \bot)) \ .$$

Proof. It is easy to see that $FPHP_n^m$ formula is totally weight constrained, as every variable appears in some pigeon axiom that is constrained by the functional axioms. Also, $FPHP_n^m$ has at most 2 negative literals in each clause and hence we have that $Sp_{PCR}(FPHP_n^m \vdash \bot) = \Theta(Sp_{PC}(FPHP_n^m \vdash \bot))$. □

Actually, we can say even more about the space complexity of the functional pigeonhole principle formulas. In [FLN+12], the authors prove that the PCR space complexity of $FPHP_n^m$ is equal (up to constant factors) to the PCR space complexity of the extended formula $\widetilde{FPHP}_n^m$, where $\widetilde{FPHP}_n^m$ is the canonical equivalent 3-CNF version[7] of the formula $FPHP_n^m$. Hence, we have that the PC space complexity lower bound for $FPHP_n^m$ would actually lower bound the PCR space complexity of $\widetilde{FPHP}_n^m$ and give us the first PCR space lower bound for some family of 3-CNF formulas.

This holds in greater generality for totally weight constrained formulas that also fulfill the following technical condition: F is a *weight-constrained* CNF formula if for each clause $l_1 \vee l_2 \vee \ldots \vee l_m$ of F with more than three literals, the formula also contains clauses $\neg l_i \vee \neg l_j$ for all $1 \leq i < j \leq m$. We stress the fact that the conditions of being weight-constrained and totally weight constrained are incomparable.

Corollary 67. *For a simultaneously weight-constrained and a totally weight constrained formula F, where each clause has a costant number of negative literals, it holds that*

$$Sp_{PCR}(\widetilde{F} \vdash \bot) = \Theta(Sp_{PCR}(F \vdash \bot)) = \Theta(Sp_{PC}(F \vdash \bot)) \ .$$

9 Concluding Remarks

In this paper, following up on recent work in [BNT13, BG13, FLN+12, HN12], we report further progress on understanding space complexity in polynomial calculus and how the space measure is related to size and degree. Specifically, we separate size

[7] We substitute every clause $l_1 \vee l_2 \vee \ldots \vee l_k$, which has more than three literals, with the formula $(l_1 \vee y_1) \wedge (\neg y_1 \vee l_2 \vee y_2) \wedge \ldots \wedge (\neg y_{i-1} \vee l_i \vee y_i) \wedge \ldots \wedge (\neg y_{k-1} \vee l_k)$ where for each substituted clause all variables y_i are new. The substituted formula is a 3-CNF and it is satisfiable if and only if the original one is. It is also easy to deduce the original clause from the substituting formula.

and degree from space, and provide some circumstantial evidence for the conjecture that degree might be a lower bound on space in PC/PCR. We also prove space lower bounds for a large class of Tseitin formulas, a well-studied formula family for which nothing was previously known regarding PCR space.

We believe that our lower bounds for Tseitin formulas over random graphs are *not* optimal, however. And for the functional pigeonhole principle, we show that the technical tools developed in [BG13] cannot prove any non-constant PCR space lower bounds. Although we have not been able to prove this, we believe that similar impossibility results should hold also for ordering principle formulas and for the canonical 3-CNF version of the pigeonhole principle. Since all of these formulas require large degree in PCR and large space in resolution, it is natural to suspect that they should be hard for PCR space as well. The fact that arguments along the lines of [BG13] do not seem to be able to establish this suggests that we are still far from a combinatorial characterization of degree analogous to the characterization of resolution width in [AD08].

It thus remains a major open problem to understand the relation between degree and space in PC/PCR, and in particular whether degree is a lower bound on space or not (or whether it even holds that resolution width provides a lower bound on PCR space).

Also, our separations of size and degree on the one hand and space on the other depend on the characteristic of the underlying field, in that the characteristic must be chosen first and the formula family exhibiting the separation works only for this specific characteristic. It would be satisfying to find formulas that provide such separations regardless of characteristic. Natural candidates are (various flavours of) ordering principle formulas or onto function pigeon principle formulas, or, for potentially even stronger separations, pebbling formulas.

Finally, an intriguing question is how (monomial) space in PC/PCR is related to (clause) space in resolution. There are separations known for size versus length and degree versus width, and it would seem reasonable to expect that PCR should be strictly stronger than resolution also with respect to space, but this is completely open.[8] The flipside of this question is to what extent space lower bound techniques for resolution carry over to PC/PCR. Since so far we do not know of any counter-examples, it is natural to ask, for instance, whether *semiwide* CNF formulas as defined in [ABRW02] have high space complexity not only in resolution but also in PCR.

Acknowledgements

The authors wish to thank Ilario Bonacina and Nicola Galesi for numerous and very useful discussions.

[8]For completeness, we mention that there is a very weak (constant-factor) separation in [ABRW02], but it crucially depends on a somewhat artificial definition of space where monomials are *not* counted with repetitions.

The research of the first author has received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 238381. Part of the work of the first author was performed while visiting KTH Royal Institute of Technology. The other authors were funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 279611. The fourth author was also supported by Swedish Research Council grants 621-2010-4797 and 621-2012-5645.

A PCR Space Lower Bounds from Extendible Families

For the sake of self-containment, in this appendix we give an exposition of the Bonacina-Galesi framework [BG13] for proving space lower bounds in Polynomial Calculus. We show how the existence of a r-extendible family for a large value of r implies such bounds. This framework can actually prove space lower bounds for a proof system that it stronger than PC or PCR.

Definition 68 (Functional Calculus (FC)). A *functional calculus* configuration is a set of arbitrary Boolean functions over Boolean variables. There is a single derivation rule, *semantic implication*, where g can be inferred from $f_1, \ldots, f_n$ if every assignment that satisfies $f_1 \wedge \cdots \wedge f_n$ also satisfies g.

Verifying a proof in FC is coNP-complete, and so FC is not a proof system in the sense of Cook and Reckhow [CR79] unless coNP = NP.

There are many different circuit representations of the same Boolean function, so we need to choose a minimal representation in order to define clause space.

Definition 69. Let $\mathbb{P}$ be a FC configuration. A set of monomials $U = \{m_1, \ldots, m_s\}$ *defines* $\mathbb{P}$ if for every function $f \in \mathbb{P}$ there is a function g such that $g(m_1, \ldots, m_s) \equiv f(x_1, \ldots, x_n)$. The *monomial space* of $\mathbb{P}$ is the minimum size of a defining set of monomials.

We can interpret polynomials in PCR as Boolean functions if we project them to the Boolean ring $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots] / \operatorname{Span}\left(x^2 - x, 1 - x - \overline{x}, \overline{x}^2 - \overline{x}, y^2 - y, \ldots\right)$. Furthermore, the set of monomials in a PCR configuration counted without repetitions is a defining set of monomials for a FC configuration. Therefore we can view every proof in PCR as a proof in FC that uses at most the same space. In particular, $Sp_{\mathcal{FC}}(F \vdash \bot) \leq Sp_{\mathcal{PCR}}(F \vdash \bot)$.

We now prove Theorem 6, following Bonacina and Galesi [BG13]. The general plan of the proof is to consider a FC derivation of a formula F in small space, and show that every configuration arising in the derivation is satisfiable. Since a refutation ends with an unsatisfiable configuration, the derivation is not a refutation.

In order to show that every configuration arising in the derivation is satisfiable, we maintain a satisfiability witness, in the form of a structured set of assignments together with a CNF formula. The following definition captures the sense in which a satisfiability witness guarantees that a board configuration is satisfiable. Fix a set

of variables V and consider partitions and total assignments with respect to this set. Recall that a total assignment assigns a value to *each* variable in V.

Definition 70. Let $(\mathcal{Q}, \mathcal{H})$ be a structured set of assignments, G be a CNF formula, and $\mathbb{P}$ be a set of Boolean functions. We write $G \models_{(\mathcal{Q},\mathcal{H})} \mathbb{P}$ if every total assignment that extends some partial assignment in $\mathcal{H}$ and satisfies G also satisfies $\mathbb{P}$.

In the proof, $\mathbb{P}$ is the contents of the board at a given point in the FC refutation, and $(\mathcal{Q}, \mathcal{H}), G$ together form a satisfiability witness. The CNF G is composed of two parts: a satisfiable subset $F' \subset F$, which could be empty, and a 2-CNF M with a very specific form given by the following definition.

Definition 71. Let M be a 2-CNF formula over the variables V. We say that M is a *transversal* of a partial partition $\mathcal{Q}$ defined on V if M mentions exactly one variable from each block $Q_i \in \mathcal{Q}$. (In particular, $|\mathcal{Q}|$ must be even and the number of clauses in M is $|\mathcal{Q}|/2$.)

A transversal CNF formula is always satisfiable, and so for $F' = \emptyset$, any board configuration $\mathbb{P}$ that has a satisfiability witness of this form must in fact be satisfiable. To handle an arbitrary F', we add the requirement that $(\mathcal{Q}, \mathcal{H})$ respect F'. Finally, we can formally define the concept of satisfiability witness.

Definition 72. Let $\mathbb{P}$ be a set of Boolean functions. A tuple $(F'; \mathcal{Q}, \mathcal{H}, M)$ is a *satisfiability witness* for $\mathbb{P}$ if:

1. F' is a satisfiable CNF formula.
2. $(\mathcal{Q}, \mathcal{H})$ is a structured assignment set which respects F'.
3. M is a 2-CNF formula which is a transversal of $\mathcal{Q}$.
4. $F' \wedge M \models_{(\mathcal{Q},\mathcal{H})} \mathbb{P}$.

The *size* of a satisfiability witness $(F'; \mathcal{Q}, \mathcal{H}, M)$ is $|M|$.

We single F' out since its value is fixed while $\mathcal{Q}, \mathcal{H}, M$ are dynamic and change throughout the FC refutation.

A FC refutation is composed of three kinds of steps: axiom download, inference and erasure. It turns out that the first two steps are relatively easy to handle, as long as we maintain the invariant that the size of the satisfiability witness is $O(Sp(\mathbb{P}))$. This invariant allows us to expand the witness in order to accommodate new axioms as long as the monomial space is small enough, using the extension property of extendible families.

Erasure is more difficult, since the monomial space of the configuration could shrink, and in order to maintain the invariant, we need to shrink the witness as well. This is accomplished by the following crucial lemma, which shows that if a configuration has any satisfiability witness, then we can find another satisfiability

witness for the configuration whose size is bounded in terms of the monomial space of the configuration.

Because of the multiple representations technical issue we also need to use the locality lemma in axiom download steps, but we could omit it in a proof of a space lower bound for PCR. It is however a key piece in erasure steps.

Lemma 73 (Locality lemma). *Suppose* $(F'; \mathcal{Q}, \mathcal{H}, M)$ *is a satisfiability witness for some set of Boolean functions* $\mathbb{P}$*. There is another satisfiability witness* $(F'; \mathcal{Q}', \mathcal{H}', M')$ *for* $\mathbb{P}$ *such that* $\mathcal{Q}' \subseteq \mathcal{Q}$*,* $\mathcal{H}' = \mathcal{H}\restriction_{\mathcal{Q}'}$ *and* $|M'| \leq 2Sp(\mathbb{P})$*.*

Proof. In this proof $\mathcal{Q}[x]$ denotes the (unique) class in $\mathcal{Q}$ that contains variable x.

The starting point of the proof is understanding the relation between monomials in a defining set of monomials U of $\mathbb{P}$ and clauses in M which underlies the property $F' \wedge M \models_{(\mathcal{Q},\mathcal{H})} \mathbb{P}$. A clause $C \in M$ affects a monomial $m \in U$ whenever the two mention variables belonging to the same partition in $\mathcal{Q}$. If a clause C does not affect a monomial m, then the clause C puts no constraints on the value of m.

Formally, we construct a bipartite graph between a minimal defining set of monomials U and the set of clauses in M (which we identify with M itself). We draw an edge between $m \in U$ and $C \in M$ whenever for some $Q \in \mathcal{Q}$, both m and C mention some variable in Q.

We break U into two parts: one part which is collectively affected by a small number of clauses, and another part in which we can associate with each monomial two clauses affecting it. To this end, let U_1 be an inclusion-maximal set under the constraint $|N(U_1)| \leq 2|U_1|$, and let $U_2 = U \setminus U_1$. We partition M accordingly into $M_1 = N(U_1)$ and $M_2 = M \setminus M_1$. As a slight modification of Hall's marriage theorem shows, the maximality of U_1 implies that we can associate with each monomial in U_2 two *unique* clauses in M_2 (that is, each clause in M_2 is associated with at most one monomial). In other words, there is a double matching from U_2 to M_2. (For more details on this step, see [ABRW02, FLN$^+$12, BG13].)

We construct the new 2-CNF M' out of two parts: $M' = M_1 \cup M_2'$. The first part M_1, taken verbatim from M, takes care of U_1. The other part M_2', which we construct from the double matching, takes care of U_2.

The 2-CNF M_2' consists of one clause C_m for every monomial $m \in U_2$. In order to define C_m, let $x^a \vee y^b$ and $z^c \vee w^d$ be the two clauses in M_2 that are matched to m in the double matching. Assume without loss of generality that $m = r^e s^f m'$, where $r \in \mathcal{Q}[x]$ and $s \in \mathcal{Q}[z]$. The clause C_m is defined as $C_m = r^e \vee s^f$.

By construction, $|M'| \leq 2|U_1| + |U_2| \leq 2|U| = 2Sp(\mathbb{P})$. Having defined M', we complete the definition of the new satisfiability witness as follows. First, let $\mathcal{Q}' = \{\mathcal{Q}[x] \mid x \in \mathit{Vars}(M')\}$; this guarantees that M' is a transversal of $\mathcal{Q}'$. Observe that $\mathcal{Q}' \subseteq Q$. Second, let $\mathcal{H}' = \mathcal{H}\restriction_{\mathcal{Q}'}$. It is easy to check that $(F'; \mathcal{Q}', \mathcal{H}', M'))$ satisfies the first three properties of a satisfiability witness. It remains to prove that $F' \wedge M' \models_{(\mathcal{Q}',\mathcal{H}')} \mathbb{P}$.

In order to show that $F' \wedge M' \models_{(\mathcal{Q}',\mathcal{H}')} \mathbb{P}$, we consider an arbitrary total assignment α extending some partial assignment in $\mathcal{H}'$ and satisfying $F' \wedge M'$. We

will modify α to another total assignment β that extends some partial assignment in $\mathcal{H}$ and satisfies $F' \wedge M$, and furthermore has the property that $\beta(m) = \alpha(m)$ for every $m \in U$. By assumption, $F' \wedge M \models_{(\mathcal{Q},\mathcal{H})} \mathbb{P}$, and so $\beta(\mathbb{P}) = 0$. Since $\beta(m) = \alpha(m)$ for every $m \in U$, we conclude that $\alpha(\mathbb{P}) = 0$ as well.

We proceed to define β. For each clause $x^a \vee y^b$ in M_2, we will define β on $\mathcal{Q}[x], \mathcal{Q}[y]$ using partial assignments from $\mathcal{H}$, distinguishing two cases: the clause is matched to some monomial in U_2, or it is *unmatched*. The values of all the other variables are taken directly from α.

Suppose $m \in U_2$ is matched to the clauses $x^a \vee y^b$ and $z^c \vee w^d$ and $C_m = r^e \vee s^f$, where $\mathcal{Q}[x] = \mathcal{Q}[r]$ and $\mathcal{Q}[z] = \mathcal{Q}[s]$. (In other words, we are in exactly the same situation described above while constructing M'.) Define β on $\mathcal{Q}[x], \mathcal{Q}[y], \mathcal{Q}[z], \mathcal{Q}[w]$ using partial assignments from $\mathcal{H}$ satisfying r^e, y^b, s^f, w^d. As a result, β satisfies the clauses $x^a \vee y^b$ and $z^c \vee w^d$ and the monomial m.

For each unmatched clause $x^a \vee y^b$ in M_2, we define β on $\mathcal{Q}[x]$ and $\mathcal{Q}[y]$ using partial assignments from $\mathcal{H}$ satisfying x^a and y^b. As a result, β satisfies the clause $x^a \vee y^b$. Finally, complete the definition of β by defining $\beta(x) = \alpha(x)$ for any hitherto undefined variable x. From the construction it is clear that β extends some partial assignment in $\mathcal{H}$.

In order to complete the proof, we need to show that β satisfies $F' \wedge M$, and that β agrees with α on all the monomials in U. We start by showing that β satisfies $F' \wedge M$. By construction, β satisfies the clauses in M_2. Since β agrees with α on variables mentioned in M_1, β satisfies M_1. Finally, let $C \in F'$. Since $(\mathcal{Q}, \mathcal{H})$ respects F', either the variables in C are disjoint from $\bigcup \mathcal{Q}$, or the variables in C all belong to some $Q_i \in \mathcal{Q}$, and all assignments in the respective $H_i \in \mathcal{H}$ satisfy C. In the former case, β agrees with α on variables mentioned in C, and so β satisfies C. In the latter case, β satisfies C since β extends some partial assignment in $\mathcal{H}$.

It remains to show that $\beta(m) = \alpha(m)$ for all monomials $m \in U$. In short, this is true for monomials in U_1 since α and β agree on all the relevant variables, and for monomials in U_2 since in both assignments they are reduced to zero. We proceed to show this formally.

Suppose first that $m \in U_1$. We claim that $\alpha(v) = \beta(v)$ for all variables v mentioned in m. Indeed, if $\alpha(v) \neq \beta(v)$ then $v \in \mathcal{Q}[x]$ for some clause $C = x^a \vee y^b$ in M_2. Yet this implies that m is connected to C, contradicting the definition of M_2. We conclude that α and β agree on all variables mentioned in m, and so $\alpha(m) = \beta(m)$ in this case.

Suppose next that $m \in U_2$. We claim that $\alpha(m) = \beta(m) = 0$. Let $C_m = r^e \vee s^f$, and recall that m is of the form $m = r^e s^f m'$. Thus $\alpha(m) = 0$ since α satisfies C_m, and $\beta(m) = 0$ since it satisfies r^e and s^f by construction. □

Theorem 74 (restatement of Theorem 6 [BG13]). *Let F be a CNF formula with an r-extendible family $\mathcal{F}$ with respect to some $F' \subseteq F$. Then $Sp_{\mathcal{FC}}(F \vdash \bot) \geq r/4$.*

Proof. Let $\mathcal{F}$ be an r-extendible family with respect to some satisfiable $F' \subseteq F$. Let π be a derivation from F in space $Sp(\pi) < r/4$. We will show that $1 \notin \pi$ or, even stronger, that every configuration $\mathbb{P}_t$ appearing in π is satisfiable.

We will maintain a satisfiability witness $(F'; \mathcal{Q}_t, \mathcal{H}_t, M_t)$ for every configuration $\mathbb{P}_t$. Our satisfiability witnesses will satisfy two conditions: $(\mathcal{Q}_t, \mathcal{H}_t) \in \mathcal{F}$, and the *size bound* $|M_t| \leq 2Sp(\mathbb{P}_t)$. The existence of a satisfiability witness implies that $\mathbb{P}_t$ is satisfiable. Indeed, let $\alpha \in \mathcal{H}_t$ be some partial assignment that satisfies all the literals in M_t. Since $(\mathcal{Q}_t, \mathcal{H}_t)$ respects F', each clause in F' is either already satisfied by α or is completely disjoint from the domain of α. As F' is satisfiable, we can extend α to a total assignment β which satisfies F'. Hence, from $F' \wedge M_t \models_{(\mathcal{Q}_t, \mathcal{H}_t)} \mathbb{P}_t$ we have that β satisfies $\mathbb{P}_t$, and so $\mathbb{P}_t$ is satisfiable.

We construct the satisfiability witnesses by induction. For $t = 0$, the satisfiability witness is $(F'; \emptyset, \emptyset, \emptyset)$. For the induction step, suppose we are given a satisfiability witness $(F'; \mathcal{Q}, \mathcal{H}, M)$ for $\mathbb{P}_t$. We will construct a satisfiability witness $(F'; \mathcal{Q}', \mathcal{H}', M')$ for $\mathbb{P}_{t+1}$. To simplify the notation, let $\mathbb{P} = \mathbb{P}_t$ and $\mathbb{P}' = \mathbb{P}_{t+1}$. We distinguish three cases, which correspond to the three possible steps in the proof.

Axiom download. Let C be the downloaded clause, which we also regard as a monomial. If $C \in F'$ or every extension α of a partial assignment in $\mathcal{H}$ satisfies C, then in particular $F' \wedge M \models_{(\mathcal{Q}, \mathcal{H})} \mathbb{P} \cup \{C\} = \mathbb{P}'$, and $M' = M$, $\mathcal{Q}' = \mathcal{Q}$, $\mathcal{H}' = \mathcal{H}$ form a satisfiability witness.

Otherwise, by hypothesis $Sp(\mathbb{P}') < r/4$ and so $Sp(\mathbb{P}) < r/4 - 1$. Indeed, if U is a defining set of monomials of $\mathbb{P}$, then $U \cup \{C\}$ is a defining set of monomials of $\mathbb{P}'$. By the induction hypothesis, $|\mathcal{Q}| < r - 1$. By the extension property of extendible, there exists a structured set of assignments $(\tilde{\mathcal{Q}}, \tilde{\mathcal{H}}) \in \mathcal{F}$ such that $|\tilde{\mathcal{Q}}| < r$, $(\mathcal{Q}, \mathcal{H}) \preccurlyeq (\tilde{\mathcal{Q}}, \tilde{\mathcal{H}})$ and $\tilde{\mathcal{H}} \models C$. By assumption $\mathcal{H} \not\models C$ and so $\mathcal{Q} \neq \tilde{\mathcal{Q}}$. Let $\tilde{\mathcal{Q}} = \mathcal{Q} \cup \{Q\}$.

The assignments corresponding to Q in $\tilde{\mathcal{H}}$ will ensure that the clause C is satisfied. Since we are going to add a new clause to M', we need to come up with two new parts in $\mathcal{Q}'$, and so we repeat the process. Let D be any axiom in $F \setminus F'$ such that $\tilde{\mathcal{H}} \not\models D$; if no such axiom exists then F is satisfiable and the theorem follows vacuously. Repeat the argument above and obtain a new disjoint set Q' and a structured set of assignments $(\mathcal{Q}', \mathcal{H}') \in \mathcal{F}$.

Choose arbitrary variables $x \in Q$ and $y \in Q'$, and let $M' = M \cup \{x \vee y\}$. By construction, $(F'; \mathcal{Q}', \mathcal{H}', M')$ is a satisfiability witness for $\mathbb{P}'$.

In both cases, Lemma 73 yields another satisfiability witness $(F'; \mathcal{Q}'', \mathcal{H}'', M'')$ for $\mathbb{P}'$ satisfying the size bound and with $\mathcal{Q}'' \subseteq \mathcal{Q}'$, $\mathcal{H}'' = \mathcal{H}' \restriction_{\mathcal{Q}''}$. By the restriction property of the extendible family, we have $(\mathcal{Q}'', \mathcal{H}'') \in \mathcal{F}$.

Inference. It is enough to pick $M' = M$, $\mathcal{Q}' = \mathcal{Q}$, $\mathcal{H}' = \mathcal{H}$. The first three properties in the definition of satisfiability witness continue to hold, while the last property follows from the soundness of FC. Finally, the size bound trivially holds since $|\mathbb{P}'| \geq |\mathbb{P}|$.

Erasure. Since FC is sound, $(F'; \mathcal{Q}, \mathcal{H}, M)$ is a satisfiability witness for $\mathbb{P}'$ as well. Hence Lemma 73 furnishes us with a satisfiability witness $(F'; \mathcal{Q}', \mathcal{H}', M')$ for $\mathbb{P}'$ satisfying the size bound and with $\mathcal{Q}' \subseteq \mathcal{Q}$, $\mathcal{H}' = \mathcal{H}\restriction_{\mathcal{Q}'}$. By the restriction property of extendible, $(\mathcal{Q}', \mathcal{H}') \in \mathcal{F}$.

□

Bibliography

[ABRW02] 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*.

[AD08] 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*.

[AR03] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. *Proceedings of the Steklov Institute of Mathematics*, 242:18–35, 2003. Available at `http://people.cs.uchicago.edu/~razborov/files/misha.pdf`. Preliminary version in *FOCS '01*.

[AS00] Noga Alon and Joel Spencer. *The probabilistic method*. Wiley-Interscience, 2nd edition, 2000.

[BBI12] Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 213–232, May 2012.

[Ben09] Eli Ben-Sasson. Size-space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, May 2009. Preliminary version in *STOC '02*.

[BG03] 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*.

[BG13] Ilario Bonacina and Nicola Galesi. Pseudo-partitions, transversality and locality: A combinatorial characterization for the space measure in algebraic proof systems. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS '13)*, pages 455–472, January 2013.

[BL91] Béla Bollobás and Imre Leader. Edge-isoperimetric inequalities in the grid. *Combinatorica*, 11:299–314, 1991.

[Bla37] Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.

[BN08] 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)*, pages 709–718, October 2008.

[BN11] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, January 2011.

[BNT13] Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.

[Bol88] Béla Bollobás. The isoperimetric number of random regular graphs. *European Journal of Combinatorics*, 9:241–244, 1988.

[BS97] Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pages 203–208, July 1997.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

[CEI96] 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)*, pages 174–183, May 1996.

[CR79] Stephen A. Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979.

[CS88] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[ET01] 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*.

[FLM+13] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (Extended abstract). In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 437–448. Springer, July 2013.

[FLN+12] Yuval Filmus, Massimo Lauria, Jakob Nordström, Neil Thapen, and Noga Ron-Zewi. Space complexity in polynomial calculus (Extended abstract). In *Proceedings of the 27th Annual IEEE Conference on Computational Complexity (CCC '12)*, pages 334–344, June 2012.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[HN12] Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity (Extended abstract). In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 233–248, May 2012.

[IK10] Russell Impagliazzo and Valentine Kabanets. Constructive proofs of concentration bounds. In *Proceedings of the 13th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems and 14th International Workshop on Randomization and Computation (APPROX-RANDOM '10)*, pages 617–631, 2010.

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[JŁR00] Svante Janson, Tomasz Łuczak, and Andrzej Ruciński. *Random graphs*. Wiley-Interscience, 2000.

[KW01] Jeong Han Kim and Nicholas C. Wormald. Random matchings which induce Hamilton cycles, and hamiltonian decompositions of random regular graphs. *Journal of Combinatorial Theory B*, 81:20–44, 2001.

[Mor94] M. Morgenstern. Existence and explicit constructions of $q + 1$ regular Ramanujan graphs for every prime power q. *Journal of Combinatorial Theory, Series B*, 62(1):44–62, 1994.

[MS96] João P. Marques-Silva and Karem A. Sakallah. GRASP—a new search algorithm for satisfiability. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '96)*, pages 220–227, November 1996.

[PS97] Alessandro Panconesi and Aravind Srinivasan. Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997.

[Raz98] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, December 1998.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

[Wor99] Nicholas C. Wormald. Models of random regular graphs. In *Surveys in Combinatorics*, pages 239–298. Cambridge University Press, 1999.

B

From Small Space to Small Width in Resolution*

Yuval Filmus[1], Massimo Lauria[2], Mladen Mikša[2], Jakob Nordström[2], and Marc Vinyals[2]

[1]Institute for Advanced Study
[2]KTH Royal Institute of Technology

Abstract

In 2003, Atserias and Dalmau resolved a major open question about the resolution proof system by establishing that the space complexity of a CNF formula is always an upper bound on the width needed to refute the formula. Their proof is beautiful but uses a nonconstructive argument based on Ehrenfeucht-–Fraïssé games. We give an alternative, more explicit, proof that works by simple syntactic manipulations of resolution refutations. As a by-product, we develop a "black-box" technique for proving space lower bounds via a "static" complexity measure that works against any resolution refutation—previous techniques have been inherently adaptive. We conclude by showing that the related question for polynomial calculus (i.e., whether space is an upper bound on degree) seems unlikely to be resolvable by similar methods.

1 Introduction

A *resolution proof* for, or *resolution refutation* of, an unsatisfiable formula F in conjunctive normal form (CNF) is a sequence of disjunctive clauses $(C_1, C_2, \ldots, C_\tau)$, where every clause C_t is either a member of F or is logically implied by two previous clauses, and where the final clause is the contradictory empty clause $\bot$ containing no literals. Resolution is arguably the most well-studied proof system in propositional proof complexity, and has served as a natural starting point in the quest to prove lower bounds for increasingly stronger proof systems on *proof length/size* (which for resolution is the number of clauses in a proof).

*This is a modified version of the paper [FLM+15], which appeared in *ACM Transactions on Computational Logic*. The paper was first presented in the *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS '14) [FLM+14].*

Resolution is also intimately connected to SAT solving in that it lies at the foundation of state-of-the-art SAT solvers using so-called conflict-driven clause learning (CDCL). This connection has motivated the study of *proof space* as a second interesting complexity measure for resolution. The space usage at some step t in a proof is measured as the number of clauses occurring before C_t that will be used to derive clauses after C_t, and the space of a proof is obtained by taking the maximum over all steps t.

For both of these complexity measures, it turns out that a key role is played by the auxiliary measure of *width*, i.e., the size of a largest clause in the proof. In a celebrated result, Ben-Sasson and Wigderson [BW01] showed that there are short resolution refutations of a formula if and only if there are also (reasonably) narrow ones, and almost all known lower bounds on resolution length can be (re)derived using this connection. In 2003, Atserias and Dalmau (journal version in [AD08]) established that width also provides lower bounds on space, resolving a problem that had been open since the study of space complexity of propositional proofs was initiated in the late 1990s in [ABRW02, ET01]. This means that for space also, almost all known lower bounds can be rederived by using width lower bounds and appealing to [AD08]. This is not a two-way connection, however, in that formulas of almost worst-case space complexity may require only constant width, as shown in [BN08].

Our Contributions

The starting point of our work is the lower bound on space in terms of width in [AD08]. This is a very elegant but also indirect proof in that it translates the whole problem to Ehrenfeucht–Fraïssé games in finite model theory, and shows that resolution space and width correspond to strategies for two opposite players in such games. Unfortunately, this also means that one obtains essentially no insight into what is happening on the proof complexity side (other than that the bound on space in terms of width is true). It has remained an open problem to give a more explicit, proof complexity theoretic argument.

In this paper, we give a purely combinatorial proof in terms of simple syntactic manipulations of resolution refutations. To summarize in one sentence, we study the conjunctions of clauses in memory at each time step in a small-space refutation, negate these conjunctions and then expand them to conjunctive normal form again, and finally argue that the new sets of clauses listed in reverse order (essentially) constitute a small-width refutation of the same formula.[1]

This new, simple proof also allows us to obtain a new technique for proving space lower bounds. This approach is reminiscent of [BW01] in that one defines a static "progress measure" on refutations and argues that when a refutation has made substantial progress it must have high complexity with respect to the

[1] We recently learned that a similar proof, though phrased in a slightly different language, was obtained independently by Razborov [Raz14].

proof complexity measure under study. Previous lower bounds on space have been inherently adaptive and in that sense less explicit.

One important motivation for our work was the hope that a simplified proof of the space-width inequality would serve as a stepping stone to resolving the analogous question for the polynomial calculus proof system. Here the width of clauses corresponds to the *degree* of polynomials, space is measured as the total number of monomials of all polynomials currently in memory, and the problem is to determine whether space and degree in polynomial calculus are related in the same way as are space and width in resolution. A possible approach for attacking this question was proposed in [BG13]. In [FLM$^+$13] we obtained a result analogous to [BN08] that there are formulas of worst-case space complexity that require only constant degree. The question of whether degree lower bounds imply space lower bounds remains open, however, and other results in [FLM$^+$13] can be interpreted as implying that the techniques in [BG13] probably are not sufficient to resolve this question. Unfortunately, as discussed towards the end of this paper we also show that it appears unlikely that this problem can be addressed by methods similar to our proof of the corresponding inequality for resolution.

Outline of This Paper

The rest of this paper is organized as follows. After some brief preliminaries in Section 2, we present the new proof of the space-width inequality in [AD08] in Section 3. In Section 4 we showcase the new technique for space lower bounds by studying so-called Tseitin formulas. Section 5 explains why we believe it is unlikely that our methods will extend to polynomial calculus. Some concluding remarks are given in Section 6.

2 Preliminaries

Let us start with a brief review of the preliminaries. The following material is standard and can be found, e.g., in the survey [Nor13].

A *literal* over a Boolean variable x is either the variable x itself (a *positive literal*) or its negation that is denoted either as $\neg x$ or as $\overline{x}$ (a *negative literal*). We define $\overline{\overline{x}} = x$. A *clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of literals and a *term* $T = a_1 \wedge \cdots \wedge a_k$ is a conjunction of literals. We denote the empty clause by $\bot$ and the empty term by $\emptyset$. The logical negation of a clause $C = a_1 \vee \cdots \vee a_k$ is the term $\overline{a}_1 \wedge \cdots \wedge \overline{a}_k$ that consists of the negations of the literals in the clause. We will sometimes use the notation $\neg C$ or $\overline{C}$ for the term corresponding to the negation of a clause C and $\neg T$ or $\overline{T}$ for the clause negating a term T. A clause (term) is *trivial* if it contains both a variable and its negation. For the proof systems we study, trivial clauses and terms can always be eliminated without any loss of generality.

A clause C' *subsumes* another clause C if every literal from C' also appears in C. A *k-clause (k-term)* is a clause (term) that contains at most k literals. A

CNF formula $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses, and a *DNF formula* $F = T_1 \vee \cdots \vee T_m$ is a disjunction of terms. A *k-CNF formula (k-DNF formula)* is a CNF formula (DNF formula) consisting of k-clauses (k-terms). We think of clauses, terms, and CNF formulas as sets: the order of elements is irrelevant and there are no repetitions. We also assume that CNF formulas are non-trivial in the sense that they do not contain the contradictory empty clause (this is just for technical simplicity to avoid a pathological corner case).

Let us next describe a slight generalization of the resolution proof system by Krajíček [Kra01], who introduced the family of *r-DNF resolution* proof systems, denoted $\mathcal{R}(r)$, as an intermediate step between resolution and depth-2 Frege systems. An *r-DNF resolution configuration* $\mathbb{C}$ is a set of r-DNF formulas. An *r-DNF resolution refutation* of a CNF formula F is a sequence of configurations $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$ such that $\mathbb{C}_0 = \emptyset$, $\bot \in \mathbb{C}_\tau$, and for $1 \leq t \leq \tau$ we obtain $\mathbb{C}_t$ from $\mathbb{C}_{t-1}$ by one of the following steps:

Axiom download $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{A\}$, where $A \notin \mathbb{C}_{t-1}$ is a clause in F (sometimes referred to as an *axiom clause*).

Inference $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{D\}$, where $D \notin \mathbb{C}_{t-1}$ is inferred by one of the following rules (where G, H denote r-DNF formulas, T, T' denote r-terms, and $a_1, \ldots, a_r$ denote literals):

r**-cut** $\dfrac{(a_1 \wedge \cdots \wedge a_{r'}) \vee G \qquad \overline{a}_1 \vee \cdots \vee \overline{a}_{r'} \vee H}{G \vee H}$, where $r' \leq r$.

$\wedge$**-introduction** $\dfrac{G \vee T \qquad G \vee T'}{G \vee (T \wedge T')}$, as long as $|T \cup T'| \leq r$.

$\wedge$**-elimination** $\dfrac{G \vee T}{G \vee T'}$ for any non-empty $T' \subseteq T$.

Weakening $\dfrac{G}{G \vee H}$ for any r-DNF formula H.

Erasure $\mathbb{C}_t = \mathbb{C}_{t-1} \setminus \{D\}$ for $D \in \mathbb{C}_{t-1}$.

For $r = 1$ we obtain the standard *resolution* proof system. In this case the only nontrivial inference rules are weakening and r-cut, where the former can be eliminated without loss of generality (but is sometimes convenient to have for technical purposes) and the latter simplifies to the *resolution rule*

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D} \ . \tag{2.1}$$

We identify a resolution configuration $\mathbb{C}$ with the CNF formula $\bigwedge_{C \in \mathbb{C}} C$.

The *length* $L(\pi)$ of an r-DNF resolution refutation π is the number of download and inference steps, and the *space* $Sp(\pi)$ is the maximal number of r-DNF formulas in any configuration in π. We define the length $L_{\mathcal{R}(r)}(F \vdash \bot)$ and the space

$Sp_{\mathcal{R}(r)}(F \vdash \bot)$ of refuting a formula F in r-DNF resolution by taking the minimum over all refutations F with respect to the relevant measure. We drop the proof system $\mathcal{R}(r)$ from this notation when it is clear from context.

For the resolution proof system, we also define the *width* $W(\pi)$ of a resolution refutation π as the size of a largest clause in π, and taking the minimum over all resolution refutations we obtain the width $W(F \vdash \bot)$ of refuting F. We remark that in the context of resolution the space measure defined above is sometimes referred to as *clause space* to distinguish it from other space measures studied for this proof system.

3 From Space to Width

In this section we present our new combinatorial proof that width is a lower bound for clause space in resolution. The formal statement of the theorem is as follows.

Theorem 1 ([AD08]). *Let F be a k-CNF formula and let $\pi : F \vdash \bot$ be a resolution refutation in clause space $Sp(\pi) = s$. Then there is a resolution refutation π' of F in width $W(\pi') \leq s + k - 3$.*

The proof idea is to take the refutation π in space s, negate the configurations one by one, rewrite them as equivalent sets of disjunctive clauses, and list these sets of clauses in reverse order. This forms the skeleton of the new refutation, where all clauses have width at most s. To see this, note that each configuration in the original refutation is the conjunction of at most s clauses. Therefore, the negation of such a configuration is a disjunction of at most s terms, which is equivalent (using distributivity) to a conjunction of clauses of width at most s. To obtain a legal resolution refutation, we need to fill in the gaps between adjacent sets of clauses. In this process the width might increase slightly from s to $s + k - 3$.

Before presenting the full proof, we need some technical results. We start by giving a formal definition of what we mean by a *negated configuration*.

Definition 2. The *negated configuration* $\mathrm{neg}(\mathbb{C})$ of a clause configuration $\mathbb{C}$ is defined inductively as follows:

- $\mathrm{neg}(\emptyset) = \{\bot\}$,
- $\mathrm{neg}(\mathbb{C} \cup \{C\}) = \{D \vee \overline{a} \mid D \in \mathrm{neg}(\mathbb{C})\,; a \in C \setminus D\,; \nexists B \in \mathrm{neg}(\mathbb{C}) \text{ s.t. } B \vee \overline{a} \subsetneq D \vee \overline{a}\}$.

Note that this definition makes sure that $\mathrm{neg}(\mathbb{C})$ will not contain any trivial or subsumed clauses, and it also yields that $\mathrm{neg}(\{\bot\}) = \emptyset$.

Each clause of the original configuration $\mathbb{C}$ contributes at most one literal to each clause of the negated configuration $\mathrm{neg}(\mathbb{C})$. This implies an upper bound on the width of the clauses in $\mathrm{neg}(\mathbb{C})$ as stated next.

Observation 3. *The width of any clause in the negated configuration* $\mathrm{neg}(\mathbb{C})$ *is at most* $Sp(\mathbb{C}_t) = |\mathbb{C}|$.

In our proofs we will use a different characterization of negated configurations that is easier to work with. We state this characterization as a formal proposition.

Proposition 4. *The negated configuration* $\mathrm{neg}(\mathbb{C})$ *is the set of all minimal (non-trivial) clauses* C *such that* $\neg C$ *implies the configuration* $\mathbb{C}$*. That is,*

$$\mathrm{neg}(\mathbb{C}) = \{C \mid \neg C \vDash \mathbb{C} \text{ and for every } C' \subsetneq C \text{ it holds that } \neg C' \nvDash \mathbb{C}\} \ .$$

Proof. Let us fix the configuration $\mathbb{C}$ and let $\mathbb{D}$ denote the set of all minimal clauses implying $\mathbb{C}$. We prove that for each clause $C \in \mathrm{neg}(\mathbb{C})$ there is a clause $C' \in \mathbb{D}$ such that $C' \subseteq C$ and vice versa. The proposition then follows because by definition neither $\mathbb{D}$ nor $\mathrm{neg}(\mathbb{C})$ contains subsumed clauses.

First, let $C \in \mathrm{neg}(\mathbb{C})$. By the definition of $\mathrm{neg}(\mathbb{C})$ we know that for every clause $D \in \mathbb{C}$ the clause C contains the negation of some literal from D. Hence, $\neg C$ implies $\mathbb{C}$ as it is a conjunction of literals from each clause in $\mathbb{C}$. By taking a minimal clause $C' \subseteq C$ such that $\neg C' \vDash \mathbb{C}$ we have that $C' \in \mathbb{D}$.

In the opposite direction, we want to show for any $C \in \mathbb{D}$ that C must contain a negation of some literal in D for every clause $D \in \mathbb{C}$. Assume for the sake of contradiction that $D \in \mathbb{C}$ is a clause such that none of its literals has a negation appearing in C. Let α be a total truth value assignment that satisfies $\neg C$ (such an assignment exists because C is non-trivial). By assumption, flipping the variables in α so that they falsify D cannot falsify $\neg C$. Therefore, we can find an assignment that satisfies $\neg C$ but falsifies $D \in \mathbb{C}$, which contradicts the definition of $\mathbb{D}$. Hence, C must contain a negation of some literal in D for every $D \in \mathbb{C}$ and by the definition of $\mathrm{neg}(\mathbb{C})$ there is a $C' \in \mathrm{neg}(\mathbb{C})$ such that $C' \subseteq C$. □

The following observation, which formalizes the main idea behind the concept of negated configurations, is an immediate consequence of Proposition 4.

Observation 5. *An assignment satisfies a clause configuration* $\mathbb{C}$ *if and only if it falsifies the negated clause configuration* $\mathrm{neg}(\mathbb{C})$*. That is,* $\mathbb{C}$ *is logically equivalent to* $\neg\mathrm{neg}(\mathbb{C})$.

Recall that we want to take a resolution refutation $\pi = (\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau)$ and argue that if π has small space complexity, then the reversed sequence of negated configurations $\pi' = (\mathrm{neg}(\mathbb{C}_\tau), \mathrm{neg}(\mathbb{C}_{\tau-1}), \ldots, \mathrm{neg}(\mathbb{C}_0))$ has small width complexity. However, as noted above π' is not necessarily a legal resolution refutation. Hence, we need to show how to derive the clauses in each configuration of the negated refutation without increasing the width by too much. We do so by a case analysis over the derivation steps in the original refutation, i.e., axiom download, clause inference, and clause erasure. The following lemma shows that for inference and erasure steps all that is needed in the reverse direction is to apply weakening.

Lemma 6. *If $\mathbb{C} \vDash \mathbb{C}'$, then for every clause $C \in \mathrm{neg}(\mathbb{C})$ there exists a clause $C' \in \mathrm{neg}(\mathbb{C}')$ such that C is a weakening of C'.*

Proof. For any clause C in $\mathrm{neg}(\mathbb{C})$ it holds by Proposition 4 that $\neg C \vDash \mathbb{C}$. Since $\mathbb{C} \vDash \mathbb{C}'$, this in turns implies that $\neg C \vDash \mathbb{C}'$. Applying Proposition 4 again, we conclude that there exists a clause $C' \subseteq C$ such that $C' \in \mathrm{neg}(\mathbb{C}')$. □

The only time in a refutation $\pi = (\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau)$ when it does not hold that $\mathbb{C}_{t-1} \vDash \mathbb{C}_t$ is when an axiom clause is downloaded at time t, and such derivation steps will require a bit more careful analysis. We provide such an analysis in the full proof of Theorem 1, which we are now ready to present.

Proof of Theorem 1. Let $\pi = (\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau)$ be a resolution refutation of F in space s. For every configuration $\mathbb{C}_t \in \pi$, let $\mathbb{D}_t = \mathrm{neg}(\mathbb{C}_t)$ denote the corresponding negated configuration. By assumption, each $\mathbb{C}_t$ contains at most s clauses, and thus Observation 3 guarantees that the clauses in $\mathbb{D}_t$ have width at most s. We need to show how to transform the sequence of clause configurations $\pi' = (\mathbb{D}_\tau, \mathbb{D}_{\tau-1}, \ldots, \mathbb{D}_0)$ into a legal resolution refutation of width at most $s + k - 3$. Let us assume first that we are dealing with CNF formulas of width $k \geq 3$, since this makes the argument slightly easier to present. At the end of the proof, we will see how argue more carefully to get rid of this assumption.

The initial configuration of the sequence π' is $\mathbb{D}_\tau$, which is the empty set by Definition 2. If $\mathbb{C}_{t+1}$ follows from $\mathbb{C}_t$ by inference or erasure, then we can derive any clause of $\mathbb{D}_t$ from a clause of $\mathbb{D}_{t+1}$ by weakening, as proven in Lemma 6. If $\mathbb{C}_{t+1}$ follows from $\mathbb{C}_t$ by axiom download, then we claim that we can derive $\mathbb{D}_t$ from $\mathbb{D}_{t+1}$ in width at most $s + k - 3$. Since the last configuration $\mathbb{D}_0$ of π' contains the empty clause $\bot$ by Definition 2, we obtain a complete resolution refutation.

Hence, all that we need to do is to analyze what happens at axioms downloads. We first observe that we can assume without loss of generality that prior to each axiom download step the space of the configuration $\mathbb{C}_t$ is at most $s - 2$. Otherwise, immediately after the axiom download step the proof π needs to erase a clause in order to maintain the space bound s. If the clause erased is the one just downloaded, we can obviously just ignore these two steps, and otherwise by reordering the axiom download and clause erasure steps we get a valid refutation of F for which it holds that $Sp(\mathbb{C}_t) \leq s - 2$.

Suppose $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{A\}$ for some axiom $A = a_1 \vee \cdots \vee a_\ell$, with $\ell \leq k$. Consider now some clause $C \in \mathbb{D}_t \setminus \mathbb{D}_{t+1}$. By Observation 3 it holds that $W(C) \leq Sp(\mathbb{C}_t) \leq s - 2$. To derive C we first download the axiom A and then show how to obtain C from the clauses in $\mathbb{D}_{t+1} \cup \{A\}$. Note that all clauses $C \vee \overline{a}_i$ for $a_i \in A$ are either contained in or are weakenings of clauses in $\mathbb{D}_{t+1}$. This follows easily from Definition 2 as adding an axiom A to the configuration $\mathbb{C}_t$ results in adding negations of literals from A to all clauses $C \in \mathbb{D}_t$. Hence, we can obtain C by the

following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{A = a_1 \vee \cdots \vee a_\ell \qquad C \vee \overline{a}_1}{C \vee a_2 \vee \cdots \vee a_\ell} \qquad C \vee \overline{a}_2}{C \vee a_3 \vee \cdots \vee a_\ell}}{\vdots}}{\dfrac{C \vee a_\ell \qquad\qquad C \vee \overline{a}_\ell}{C}} \tag{3.1}$$

When C is the empty clause, the width of this derivation is $W(A) \leq k$. Otherwise, it is upper-bounded by $W(C) + W(A) - 1 \leq s + k - 3$. Since any resolution refutation has space at least 3 (unless the formula contains the empty clause itself, but our definitions explicitly disallowed such trivial formulas), we conclude that the width of the derivation (3.1) is at most $\max(k, s + k - 3) = s + k - 3$. This in turn implies that the width of the resolution refutation constructed from π' is at most $\max(s, s + k - 3) = s + k - 3$, where the last equality follows from the assumption $k \geq 3$, and this completes the proof.

If $k < 3$, however, we have $s + k - 3 < s$, and so the argument above does not quite suffice to establish the bound claimed in the theorem. This can be taken care of by a postprocessing step as follows.[2] Recall that inference and erasure steps can only produce weakenings of clauses by Lemma 6, and axiom download steps only occur when the space is at most $s - 2$. Consider the resolution refutation constructed from π' as above, and then erase all clause configurations obtained at inference or erasure steps (i.e., via weakening) to obtain new refutation π''. It is straightforward to verify that this yields a legal refutation and that the width does not increase (since π'' contains a subset of the clauses in the previously constructed refutation). After this step the only new clauses in π'' that we need to derive at each step are those resulting from axiom downloads in the original refutation π, and as already noted the width of deriving such clauses as done in (3.1) is at most $\max(k, s + k - 3) = s + k - 3$. The theorem follows. □

The proof of Theorem 1 also works for r-DNF resolution, although the bound gets weaker as r grows. Let us state this as a theorem and sketch the proof.

Theorem 7. *Let F be a k-CNF formula and $\pi : F \vdash \bot$ be an r-DNF resolution refutation of F in space $Sp(\pi) \leq s$. Then there exists a resolution refutation π' of F in width at most $W(\pi') \leq (s-2)r + k - 1$.*

[2] Alternatively, once can simply observe directly that the theorem is true for $k < 3$. To see this, note that any unsatisfiable 1-CNF formula is refutable by resolving some literal with its negation in a single width-1 step. And any resolution derivation from a 2-CNF formula, unsatisfiable or not, has width 2, since resolving two 2-clauses always yields another 2-clause. Hence, for $k < 3$ we have that any unsatisfiable k-CNF formula can always be refuted in width at most $k \leq Sp(\pi) + k - 3$ for any refutation π (using again that $Sp(\pi) \geq 3$).

Proof sketch. We define the negated configuration $\mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C})$ of an $\mathcal{R}(r)$-configuration inductively by setting $\mathrm{neg}_{\mathcal{R}(r)}(\emptyset) = \{\bot\}$ and

$$\mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C} \cup \{C\}) = \{D \vee \overline{T} \mid D \in \mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C}); T \in C; \nexists B \in \mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C}) \text{ such that } B \vee \overline{T} \subsetneq D \vee \overline{T}; D \vee \overline{T} \text{ non-trivial}\} \quad (3.2)$$

to make sure that $\mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C})$ contains no trivial or subsumed clauses. It is easy to see that an r-DNF configuration of space s gets transformed into a resolution configuration of width at most sr. We can prove that $\mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C})$ is the set of all minimal clauses D such that $\neg D \models \mathbb{C}$ for an r-DNF configuration $\mathbb{C}$, which is an analogue of Proposition 4. The proof is essentially the same except that we reason using the terms of an r-DNF formula $C \in \mathbb{C}$ instead of its literals. With this version of Proposition 4 proved, we can immediately generalize Lemma 6 to the r-DNF case.

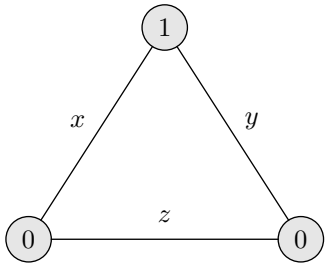
The analogue of the proof of Theorem 1 follows easily from previous observations. The inference and clause deletion steps follow by the generalized version of Lemma 6, while the case of axiom download is the same as in the original proof because axioms are clauses. Hence, running the negated r-DNF resolution refutation backwards we get a resolution refutation of F. The width of this latter refutation is at most $(s-2)r + k - 1$, as we again consider only configurations that have space equal to at most $s - 2$, and the inference steps in the case of axiom download can add at most $k - 1$ to the width of the resulting resolution refutation. When $k < 2r + 1$, an additional pruning step in which all weakenings are eliminated completes the proof. □

4 A Static Technique for Proving Space Lower Bounds

Looking at the proof complexity literature, the techniques used to prove lower bounds for resolution length and width (e.g., [BW01, CS88, Hak85, Urq87]) differ significantly from those used to prove resolution space lower bounds (e.g., [ABRW02, BG03, ET01]) in that the former are *static* or *oblivious* while the latter are *dynamic*.

Lower bounds on resolution length typically have the following general structure: if a refutation is too short, then we obtain a contradiction by applying a suitable random restriction (the length of the proof figures in by way of a union bound); so any refutation must be long. When proving lower bounds on resolution width, one defines a complexity measure and uses the properties of this measure to show that every refutation must contain a complex clause; in a second step one then argues that such a complex clause must be wide.

In contrast, most lower bound proofs for resolution space use an *adversary argument*. Assuming that the resolution derivation has small space, one constructs a satisfying assignment for each clause configuration. Such assignments are updated inductively as the derivation progresses, and one shows that the update is always possible given the assumption that the space is small. This in turn shows that the



(a) Labelled triangle graph.

$$
\begin{aligned}
&(x \vee y)\\
\wedge\ &(\overline{x} \vee \overline{y})\\
\wedge\ &(x \vee \overline{z})\\
\wedge\ &(\overline{x} \vee z)\\
\wedge\ &(y \vee \overline{z})\\
\wedge\ &(\overline{y} \vee z)
\end{aligned}
$$

(b) Corresponding Tseitin formula.

Figure 2: Example Tseitin formula.

contradictory empty clause can never be reached, implying a space lower bound on refutations. The essential feature separating this kind of proofs from the ones above is that the satisfying assignments arising during the proof *depend on the history of the derivation*; in contrast, the complexity measures in width lower bounds are defined once and for all, as are the distributions of random restrictions in length lower bounds.

In this section we present a *static* lower bound on resolution space. Our proof combines the ideas of Section 3 and the complexity measure for clauses used in [BW01]. We define a complexity measure for configurations which can be used to prove space lower bounds along the lines of the width lower bounds mentioned above.

This approach works in general in that any complexity measure for clauses can be transformed into a complexity measure for configurations. This turns many width lower bound techniques into space lower bound ones (e.g., width lower bounds for random 3-CNF formulas.) In this section we give a concrete example of this for Tseitin formulas, which are a family of CNFs encoding a specific type of systems of linear equations; see Figure 2 for illustration.

Definition 8 (Tseitin formula). Let $G = (V, E)$ be an undirected graph and $\chi : V \to \{0, 1\}$ be a function. Let us identify every edge $e \in E$ with a variable x_e, and let us write $PARITY_{v,\chi}$ to denote the canonical CNF encoding of the constraint $\sum_{e \ni v} x_e = \chi(v) \pmod 2$ for any vertex $v \in V$. Then the *Tseitin formula* over G with respect to χ is $Ts(G, \chi) = \bigwedge_{v \in V} PARITY_{v,\chi}$.

When the degree of G is bounded by d, $PARITY_{v,\chi}$ has at most 2^{d-1} clauses, all of width at most d, and hence $Ts(G, \chi)$ is a d-CNF formula with at most $2^{d-1}|V|$ clauses. We say that a set of vertices U has *odd (even) charge* if $\sum_{u \in U} \chi(u)$ is odd (even). A simple parity argument shows that when $V(G)$ has odd charge, $Ts(G, \chi)$

is unsatisfiable. On the other hand, if G is connected then for each $v \in V$ it is always possible to satisfy the constraints $PARITY_{u,\chi}$ for all $u \neq v$.

The hardness of Tseitin formulas are governed by the expansion properties of the underlying graph.

Definition 9 (Edge expander). The graph $G = (V, E)$ is an (s, δ)-*edge expander* if for every set of vertices $U \subseteq V$ such that $|U| \leq s$ it holds that $|\partial(U)| \geq \delta|U|$, where $\partial(U)$ is the set of edges of G with exactly one vertex in U.

We next present a new technique of showing that if a graph G is a good edge expander, then large space is needed to refute $Ts(G, \chi)$ in resolution. We remark that this was originally proven in [ABRW02, ET01] (and with slightly better parameters, as discussed below).

Theorem 10. *For a Tseitin formula $Ts(G, \chi)$ over a d-regular (s, δ)-edge expander G it holds that $Sp(Ts(G, \chi) \vdash \bot) \geq \delta s/d$.*

For the rest of this section we fix a particular d-regular connected graph G and a function χ with respect to which $V(G)$ has odd charge, and consider the corresponding Tseitin formula $Ts(G, \chi)$. The main tool used to prove Theorem 10 is a complexity measure for configurations. We show that if G is a good expander, then every refutation of $Ts(G, \chi)$ must have a configuration with intermediate measure. We conclude the proof by showing that the space of a configuration is at least its measure if the latter falls within a specific range of values.

We first define our configuration complexity measure for terms (i.e., configurations consisting of unit clauses), and then extend it to general configurations. In words, the term complexity measure is the smallest number of parity axioms of $Ts(G, \chi)$ that collectively contradict the term, and the configuration complexity measure is the maximum measure over all terms that imply the configuration.

Definition 11 (Configuration complexity measure). The *term complexity* $\nu(T)$ of a term T is $\nu(T) = \min\left\{|V'| : V' \subseteq V \text{ and } T \wedge \bigwedge_{v \in V'} PARITY_{v,\chi} \vDash \bot\right\}$.

The *configuration complexity measure* $\mu(\mathbb{C})$ of a resolution configuration $\mathbb{C}$ is defined as $\mu(\mathbb{C}) = \max\{\nu(T) : T \vDash \mathbb{C}\}$. When $\mathbb{C}$ is contradictory we have $\mu(\mathbb{C}) = 0$.

Note that $\nu(T)$ is a monotone decreasing function, since $T \subseteq T'$ implies $\nu(T) \geq \nu(T')$ by definition. Hence, we only need to look at minimal terms T for which $T \vDash \mathbb{C}$ in order to determine $\mu(\mathbb{C})$. These minimal terms are the *negations* of the clauses in $\text{neg}(\mathbb{C})$ (compare Proposition 4). We now introduce the convenient concept of *witness* for the measure.

Definition 12 (Witness of measure). A *witness* of the measure $\nu(T)$ of the term T is a set of vertices V^* for which $\nu(T) = |V^*|$ and $T \wedge \bigwedge_{v \in V^*} PARITY_{v,\chi} \vDash \bot$. Similarly, for configurations $\mathbb{C}$ a witness for $\mu(\mathbb{C})$ is a term T^* for which $\mu(\mathbb{C}) = \nu(T^*)$ and $T^* \vDash \mathbb{C}$.

There is a big gap between the measure of the initial and final configurations of a refutation, and we will see that the measure does not change much at each step. Hence, the refutation must pass through a configuration of intermediate measure. Formally, if G is connected then $\mu(\emptyset) = |V|$, because the empty term has measure $|V|$, and $\mu(\mathbb{C}) = 0$ when $\bot \in \mathbb{C}$.

To study how the measure changes during the refutation, we look separately at what happens at each type of step. As in the proof of Theorem 1, we can deal with inference and clause erasure steps together, whereas axiom downloads require more work.

Lemma 13. *If $\mathbb{C} \vDash \mathbb{C}'$ then $\mu(\mathbb{C}) \leq \mu(\mathbb{C}')$.*

Proof. Let T^* be a witness for $\mu(\mathbb{C})$. Then, $T^* \vDash \mathbb{C}$ and, hence, we also have $T^* \vDash \mathbb{C}'$. Therefore, $\mu(\mathbb{C}') \geq \nu(T^*)$, because $\mu(\mathbb{C}')$ is equal to the maximum value of $\nu(T)$ for terms T implying $\mathbb{C}'$. As $\nu(T^*)$ is equal to $\mu(\mathbb{C})$, the bound $\mu(\mathbb{C}') \geq \mu(\mathbb{C})$ follows. □

Lemma 14. *For a clause A in $Ts(G, \chi)$ and a graph G of bounded degree d, if $\mathbb{C}' = \mathbb{C} \cup \{A\}$ then $d \cdot \mu(\mathbb{C}') + 1 \geq \mu(\mathbb{C})$.*

Proof. Fix a witness T^* for $\mu(\mathbb{C})$. Since $\mu(\mathbb{C}) = \nu(T^*)$, to prove the lemma we need to upper-bound the value $\nu(T^*)$ by $d \cdot \mu(\mathbb{C}') + 1$.

For any literal a in A, we know that $T^* \wedge a$ implies $\mathbb{C}'$ because T^* implies $\mathbb{C}$ and a implies A. Hence, it holds that $\mu(\mathbb{C}') \geq \nu(T^* \wedge a)$, and so it will be sufficient to relate $\nu(T^*)$ to the values $\nu(T^* \wedge a)$. To this end, we look at the set of vertices $V^* = \bigcup_{a \in A} V_a \cup \{v_A\}$, where each V_a is a witness for the corresponding measure $\nu(T^* \wedge a)$, and v_A is the vertex such that $A \in PARITY_{v_A,\chi}$. Note that by definition it holds that $|V_a| = \nu(T^* \wedge a)$ for every $a \in A$, and also that $|V^*| \leq 1 + \sum_{a \in A} |V_a|$, which sum can in turn be bounded by $d \cdot \mu(\mathbb{C}') + 1$ because A has at most d literals.

We conclude the proof by showing that $T^* \wedge \bigwedge_{v \in V^*} PARITY_{v,\chi} \vDash \bot$, which establishes that $\nu(T^*) \leq |V^*|$. The implication holds because any assignment either falsifies the clause A, and so falsifies $PARITY_{v_A,\chi}$, or satisfies one of the literals $a \in A$. But then we have as a subformula $T^* \wedge \bigwedge_{v \in V_a} PARITY_{v,\chi}$, which is unsatisfiable by the definition of V_a when a is true. The bound $\nu(T^*) \leq |V^*|$ then follows, and so $\mu(\mathbb{C}) \leq |V^*| \leq d \cdot \mu(\mathbb{C}') + 1$. □

The preceding results imply that every resolution refutation of the Tseitin formula has a configuration of intermediate complexity. This holds because every refutation starts with a configuration of measure $|V|$ and needs to reach the configuration of measure 0, as noted above, while at each step the measure drops by a factor of at most $1/d$ by the lemmas we just proved. Let us state this formally as a corollary.

Corollary 15. *For any resolution refutation π of a Tseitin formula $Ts(G, \chi)$ over a connected graph G of bounded degree d and any positive integer $r \leq |V|$ there exists*

a configuration $\mathbb{C} \in \pi$ such that the configuration complexity measure is bounded by $r/d \leq \mu(\mathbb{C}) \leq r$.

It remains to show that a configuration having intermediate measure must also have large space. This part of the proof relies on the graph being an expander.

Lemma 16. *Let G be an (s, δ)-edge expander graph. For every configuration $\mathbb{C}$ satisfying $\mu(\mathbb{C}) \leq s$ it holds that $Sp(\mathbb{C}) \geq \delta \cdot \mu(\mathbb{C})$.*

Proof. To prove the lemma, we lower-bound the size of a minimal witness T^* for $\mu(\mathbb{C})$ and then use the bound $Sp(\mathbb{C}) \geq |T^*|$. This inequality follows by noting that at most one literal per clause in $\mathbb{C}$ is needed in the implying term T^*.

Fix T^* to be a minimal witness for $\mu(\mathbb{C})$ and let V^* be a witness for $\nu(T^*)$. Note that $|V^*| = \mu(\mathbb{C})$. We prove that T^* must contain a variable for every edge in $\partial(V^*)$. Towards contradiction, assume that T^* does not contain some x_e for an edge e in $\partial(V^*)$, and let v_e be a vertex in V^* incident to e. Let α be an assignment that satisfies $T^* \wedge \bigwedge_{v \in V^* \setminus \{v_e\}} PARITY_{v,\chi}$. Such an assignment must exist as otherwise V^* would not be a witness for $\nu(T^*)$. We can modify α by changing the value of x_e so that $PARITY_{v_e,\chi}$ is satisfied. By the assumption, the new assignment α' still satisfies T^* and $\bigwedge_{v \in V^* \setminus \{v_e\}} PARITY_{v,\chi}$ as neither contains the variable x_e. Thus, we have found an assignment satisfying $T^* \wedge \bigwedge_{v \in V^*} PARITY_{v,\chi}$, which is a contradiction.

Hence, the term T^* contains a variable for every edge in $\partial(V^*)$. Since G is an (s, δ)-edge expander and $|V^*| \leq s$, the term T^* contains at least $\delta \cdot |V^*|$ variables. From the inequality $Sp(\mathbb{C}) \geq |T^*|$ and the fact that $|V^*| = \mu(\mathbb{C})$ it follows that $Sp(\mathbb{C}) \geq \delta \cdot \mu(\mathbb{C})$ when $\mu(\mathbb{C}) \leq s$. □

The preceding lemma and Corollary 15 together imply Theorem 10, because by Corollary 15 there is a configuration with measure between s/d and s, and this configuration has space at least $\delta s/d$ by Lemma 16.

We want to point out that Theorem 10 gives inferior results compared to a direct application of Theorem 1 to known width lower bounds. The bounds that we get are worse by a multiplicative factor of $1/d$. One might have hoped to remove this multiplicative factor by improving the bound in Lemma 14, but this is not possible because this lemma is tight.

To see this, suppose that the graph G is a d-star: it consists of a center v which is connected to d petals $u_1, \ldots, u_d$ by the edges $e_1, \ldots, e_d$, the charge of the center is $\chi(v) = 1$, and the charges of the petals are $\chi(u_1) = \cdots = \chi(u_d) = 0$. Let $A \in PARITY_{v,\chi}$ be the axiom $A = x_{e_1} \vee \cdots \vee x_{e_d}$. Taking $\mathbb{C} = \emptyset$ and $\mathbb{C}' = \{A\}$, we have that $\mu(\mathbb{C}) = d + 1$ while $\mu(\mathbb{C}') = 1$. The latter equality holds because every minimal term implying A is of the form x_{e_i}, a term which is contradicted by the single axiom $\overline{x}_{e_i} \in PARITY_{u_i,\chi}$. Hence, we have an example where $d \cdot \mu(\mathbb{C}') + 1 = \mu(\mathbb{C})$, which shows that Lemma 14 is tight.

5 From Small Space to Small Degree in Polynomial Calculus?

An intriguing question is whether an analogue of the bound in Theorem 1 holds also for the stronger algebraic proof system *polynomial calculus* introduced in [CEI96]. In this context, it is more relevant to discuss the variant of this system presented in [ABRW02], known as *polynomial calculus (with) resolution* or *PCR*, which we briefly describe below.

In a PCR derivation, configurations are sets of polynomials in $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$, where x and $\overline{x}$ are different formal variables. Each polynomial P appearing in a configuration corresponds to the assertion $P = 0$. The proof system contains axioms $x^2 - x$ and $x + \overline{x} - 1$, which restrict the values of the variables to $\{0, 1\}$, and enforce the complementarity of x and $\overline{x}$. A literal has truth value *true* if it is equal to 0, and truth value *false* if it is equal to 1. Each clause C is translated to a monomial m with the property that $m = 0$ if and only if C is satisfied. For example, the clause $x \vee y \vee \overline{z}$ is translated to the monomial $xy\overline{z}$. There are two inference rules, *linear combination* $\frac{p \quad q}{\alpha p + \beta q}$ and *multiplication* $\frac{p}{xp}$, where p and q are (previously derived) polynomials, the coefficients α, β are elements of $\mathbb{F}$, and x is any variable (with or without bar). These rules are sound in the sense that if the antecedent polynomials evaluate to zero under some assignment, then so does the consequent polynomial. A CNF formula F is refuted in PCR by deriving the constant term 1 from the (monomials corresponding to the) clauses in F.

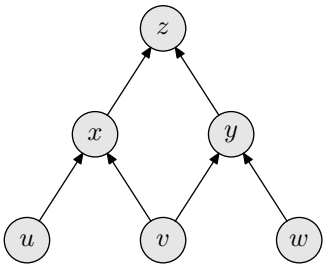
The *size*, *degree* and *monomial space* measures are analogues of length, width and clause space in resolution (counting monomials instead of clauses). PCR can simulate resolution refutations efficiently with respect to all of these measures.

Let us now discuss why the method we use to prove Theorem 1 is unlikely to generalize to PCR. An example of formulas that seem hard to deal with in this way are so-called *pebbling contradictions*, which we briefly describe next.

Pebbling contradictions are defined in terms of directed acyclic graphs (DAGs) $G = (V, E)$ with bounded fan-in, where vertices with no incoming edges are called *sources* and vertices without outgoing edges *sinks*. Assume G has a unique sink z, and associate a variable V to each vertex $v \in V$. Then the pebbling contradiction over G consists of the following clauses:

- for each source vertex s, a clause s (*source axioms*),
- for each non-source vertex v, a clause $\bigvee_{(u,v)\in E} \overline{u} \vee v$ (*pebbling axioms*),
- for the sink z, a clause $\overline{z}$ (*sink axiom*).

See Figure 4 for an illustration. Ben-Sasson [Ben09] showed that pebbling contradictions exhibit space-width trade-offs in resolution in that they can always be refuted in constant width as well as in constant space but that there are graphs for which optimizing one of these measures necessarily causes essentially worst-case linear behaviour for the other measure.



(a) Pyramid graph Π_2 of height 2.

$$
\begin{aligned}
& u \\
\wedge\ & v \\
\wedge\ & w \\
\wedge\ & (\overline{u} \vee \overline{v} \vee x) \\
\wedge\ & (\overline{v} \vee \overline{w} \vee y) \\
\wedge\ & (\overline{x} \vee \overline{y} \vee z) \\
\wedge\ & \overline{z}
\end{aligned}
$$

(b) Pebbling contradiction Peb_{Π_2}.

Figure 4: Pebbling contradiction Peb_{Π_2} for the pyramid graph Π_2 of height 2.

There are two natural ways to refute pebbling contradictions in resolution. One approach is to go "bottom-up" from sources to sinks in topological order, and derive for each vertex $v \in V(G)$ the unit clause v using the pebbling axiom for v and the unit clauses for its predecessors. When the refutation reaches z it derives a contradiction with the sink axiom $\overline{z}$. See Figure 5(a) for an example. This refutation can always be carried out in constant width but for some graphs requires large space.

The other approach is a "top-down" refutation due to [Ben09] where one starts with the sink axiom $\overline{z}$ and derives clauses of the form $\overline{v}_1 \vee \cdots \vee \overline{v}_\ell$. A new clause is derived by replacing any vertex v_i in the old one by all its predecessors, i.e., by resolving with the pebbling axiom for v_i. Since G is acyclic we can repeat this process until we get to the sources, for which the negated literals can be resolved away using source axioms. This refutation is illustrated in Figure 5(b). It is not hard to see that it can be performed in constant clause space, but it might require large width.

A careful study now reveals that the transformation of configurations in our proof of Theorem 1 maps either of the two refutations described above into the other one. Instead of providing a formal argument, we encourage the reader to compute the transformations of the refutations in Figures 5(a) and 5(b), observing that the axioms are downloaded in opposite order in the two derivations. This observation is the main reason why our proof does not seem to generalize to PCR, as we now explain.

In PCR, we can represent any conjunction of literals $a_1 \wedge \cdots \wedge a_r$ as the binomial $1 - \prod_i \overline{a}_i$. Using this encoding with the bottom-up approach yields a third refutation, which has constant space but possibly large degree: the fact that a set of vertices U "are true" can be stored as the high-degree binomial $1 - \prod_{v \in U} \overline{v}$ instead of as a collection of low-degree monomials $\{v \mid v \in U\}$. Hence, there are constant space

1.	u	Axiom
2.	v	Axiom
3.	w	Axiom
4.	$\overline{u} \vee \overline{v} \vee x$	Axiom
5.	$\overline{v} \vee x$	Res$(1, 4)$
6.	x	Res$(2, 5)$
7.	$\overline{v} \vee \overline{w} \vee y$	Axiom
8.	$\overline{w} \vee y$	Res$(2, 7)$
9.	y	Res$(3, 8)$
10.	$\overline{x} \vee \overline{y} \vee z$	Axiom
11.	$\overline{y} \vee z$	Res$(6, 10)$
12.	z	Res$(9, 11)$
13.	$\overline{z}$	Axiom
14.	$\bot$	Res$(12, 13)$

(a) Bottom-up refutation of Peb_{Π_2}.

1.	$\overline{z}$	Axiom
2.	$\overline{x} \vee \overline{y} \vee z$	Axiom
3.	$\overline{x} \vee \overline{y}$	Res$(1, 2)$
4.	$\overline{v} \vee \overline{w} \vee y$	Axiom
5.	$\overline{v} \vee \overline{w} \vee \overline{x}$	Res$(3, 4)$
6.	$\overline{u} \vee \overline{v} \vee x$	Axiom
7.	$\overline{u} \vee \overline{v} \vee \overline{w}$	Res$(5, 6)$
8.	w	Axiom
9.	$\overline{u} \vee \overline{v}$	Res$(7, 8)$
10.	v	Axiom
11.	$\overline{u}$	Res$(9, 10)$
12.	u	Axiom
13.	$\bot$	Res$(11, 12)$

(b) Top-down refutation of Peb_{Π_2}.

Figure 5: Example resolution refutations of pebbling contradiction Peb_{Π_2}.

PCR refutations of pebbling contradictions in both the bottom-up and the top-down directions. This in turn means that if our proof method were to work for PCR, we would need to find constant degree refutations in both directions. For the top-down case it seems unlikely that such a refutation exists, since clauses of the form $\bigvee_{v \in U} \overline{v}$ cannot be represented as low-degree polynomials.

6 Concluding Remarks

In this work, we present an alternative, more explicit, proof of the result by Atserias and Dalmau [AD08] that space is an upper bound on width in resolution. Our construction gives a syntactic way to convert a small-space resolution refutation into a refutation in small width. We also exhibit a new "black-box" approach for proving space lower bounds that works by defining a progress measure à la Ben-Sasson and Wigderson [BW01] and showing that when a refutation has made medium progress towards a contradiction it must be using a lot of space. We believe that these techniques shed interesting new light on resolution space complexity and hope that they will serve to increase our understanding of this notoriously tricky complexity measure.

As an example of a question about resolution space that still remains open, suppose we are given a k-CNF formula that is guaranteed to be refutable in constant space. By [AD08] it is also refutable in constant width, and a simple counting

argument then shows that exhaustive search in small width will find a polynomial-length resolution refutation. But is there any way of obtaining such a short refutation from a refutation in small space that is more explicit than doing exhaustive search? And can we obtain a short refutation without blowing up the space by more than, say, a constant factor? Known length-space trade-off results for resolution in [BBI12, BN11, BNT13, Nor09] do not answer this question as they do not apply to this range of parameters. Unfortunately, our new proof of the space-width inequality cannot be used to resolve this question either, since in the worst case the resolution refutation we obtain might be as bad as the one found by exhaustive search of small-width refutations (or even worse, due to repetition of clauses). This would seem to be inherent—a recent result [ALN14] shows that there are formulas refutable in space and width s where the shortest refutation has length $n^{\Omega(s)}$, i.e., matching the exhaustive search upper bound up to a (small) constant factor in the exponent.

An even more intriguing question is how the space and degree measures are related in polynomial calculus, as discussed in Section 5. Most relations between length, space, and width in resolution carry over with little or no modification to size, space, and degree, respectively, in polynomial calculus. So can it be that space also yields an upper bound on degree in polynomial calculus? Or could perhaps even the stronger claim hold that polynomial calculus space is an upper bound on resolution width? These questions remain wide open, but in the recent paper [FLM$^+$13] we made some limited progress by showing that if a formula requires large resolution width, then the "XORified version" of the formula requires large polynomial calculus space. We refer to the introductory section of [FLM$^+$13] for a more detailed discussion of these issues.

Acknowledgments

The authors wish to thank Albert Atserias, Ilario Bonacina, Nicola Galesi, and Li-Yang Tan for stimulating discussions on topics related to this work. We would also like to thank Alexander Razborov for sharing his proof of the theorem that space provides an upper bound on width, which is very similar to ours although expressed in a different language. Finally, we are grateful to the anonymous reviewers, who helped improve this paper considerably.

The research of the first author has received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 238381 and from the National Science Foundation under agreement no DMS-1128155. Part of the work of the first author was performed while at the University of Toronto and while visiting KTH Royal Institute of Technology supported by the foundations *Johan och Jakob Söderbergs stiftelse*, *Stiftelsen Längmanska kulturfonden*, and *Helge Ax:son Johnsons stiftelse*. The other authors were funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no. 279611. The fourth author was also supported by the Swedish Research Council grants 621-2010-4797 and 621-2012-5645. Any opinions,

findings and conclusions or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of the National Science Foundation or other funding agencies.

Bibliography

[ABRW02] 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*.

[AD08] 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*.

[ALN14] Albert Atserias, Massimo Lauria, and Jakob Nordström. Narrow proofs may be maximally long. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC '14)*, pages 286–297, June 2014.

[BBI12] Paul Beame, Chris Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC '12)*, pages 213–232, May 2012.

[Ben09] Eli Ben-Sasson. Size-space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, May 2009. Preliminary version in *STOC '02*.

[BG03] 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*.

[BG13] Ilario Bonacina and Nicola Galesi. Pseudo-partitions, transversality and locality: A combinatorial characterization for the space measure in algebraic proof systems. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science (ITCS '13)*, pages 455–472, January 2013.

[BN08] 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)*, pages 709–718, October 2008.

[BN11] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, pages 401–416, January 2011.

[BNT13] Chris Beck, Jakob Nordström, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 813–822, May 2013.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

[CEI96] 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)*, pages 174–183, May 1996.

[CS88] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[ET01] 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*.

[FLM+13] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (Extended abstract). In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP '13)*, volume 7965 of *Lecture Notes in Computer Science*, pages 437–448. Springer, July 2013.

[FLM+14] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. From small space to small width in resolution. In *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS '14)*, volume 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 300–311, March 2014.

[FLM+15] 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*.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[Kra01] Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

[Nor09] Jakob Nordström. A simplified way of proving trade-off results for resolution. *Information Processing Letters*, 109(18):1030–1035, August 2009. Preliminary version in ECCC report TR07-114, 2007.

[Nor13] Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

[Raz14] Alexander Razborov. Personal communication, 2014.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

C

Long Proofs of (Seemingly) Simple Formulas*

Mladen Mikša[1] and Jakob Nordström[1]

[1]KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

In 2010, Spence and Van Gelder presented a family of CNF formulas based on combinatorial block designs. They showed empirically that this construction yielded small instances that were orders of magnitude harder for state-of-the-art SAT solvers than other benchmarks of comparable size, but left open the problem of proving theoretical lower bounds. We establish that these formulas are exponentially hard for resolution and even for polynomial calculus, which extends resolution with algebraic reasoning. We also present updated experimental data showing that these formulas are indeed still hard for current CDCL solvers, provided that these solvers do not also reason in terms of cardinality constraints (in which case the formulas can become very easy). Somewhat intriguingly, however, the very hardest instances in practice seem to arise from so-called fixed bandwidth matrices, which are provably easy for resolution and are also simple in practice if the solver is given a hint about the right branching order to use. This raises the question of whether these formulas could be examples of SAT instances for which CDCL with current heuristics does not always search efficiently for short resolution proofs, despite the theoretical results of [Pipatsrisawat and Darwiche 2011] and [Atserias, Fichte, and Thurley 2011].

1 Introduction

Modern applied SAT solving is a true success story, with current state-of the art solvers based on *conflict-driven clause learning (CDCL)* [BS97, MS99, MMZ+01] having delivered performance improvements of orders of magnitude larger than seemed possible just 15–20 years ago. From a theoretical perspective, however, the dominance of the CDCL paradigm is somewhat surprising in that it is ultimately

*This is a slightly revised and expanded version of the paper [MN14] which appeared in *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14).*

based on the fairly weak *resolution* proof system [Bla37]. Since it is possible in principle to extract a resolution refutation of an unsatisfiable formula from the execution trace of a CDCL solver running on it, lower bounds on resolution refutation length/size yield lower bounds on the running time of any CDCL solver trying to decide this formula.[1] By now, there is a fairly extensive literature on SAT instances for which exponential lower bounds are known, imposing firm restrictions on what kind of formulas the basic CDCL approach can hope to solve.

This suggests that an interesting question might be to turn the tables and ask for *maximally hard* instances. What are the smallest CNF formulas, measured in size or number of variables, that are beyond reach of the currently best solvers? Pigeonhole principle (PHP) formulas were the first to be proven hard for resolution in the breakthrough result by Haken [Hak85], but in terms of formula size N their hardness scales only as $\exp\bigl(\Omega\bigl(\sqrt[3]{N}\bigr)\bigr)$. Two formula families with refutation length $\exp(\Omega(N))$ are Tseitin formulas[2] over so-called expander graphs and random k-CNF formulas, as shown by Urquhart [Urq87] and Chvátal and Szemerédi [CS88], respectively. The strongest lower bounds to date in terms of the explicit constant in the exponent were established recently by Beck and Impagliazzo [BI13] for formulas encoding inconsistent systems of linear equations.

Spence [Spe10] instead focused on empirical hardness and exhibited a family of 3-CNF formulas that seem practically infeasible even for very small instances (around 100 variables). These formulas can be briefly described as follows. Fix a set of $4n+1$ variables. Randomly partition the variables into groups of 4 plus one group of 5. For each 4-group, write down the natural 3-CNF formula encoding the *positive cardinality constraint* that at least 2 variables must be true, and for the 5-group encode that a strict majority of 3 variables must be true. Do a second random variable partition into 4-groups plus one 5-group, but now encode *negative cardinality constraints* that the number of false variables is at least 2 and 3, respectively. By a counting argument, the CNF formula consisting of the conjunction of all these clauses must be unsatisfiable. Although [Spe10] does not present any theoretical analysis, these formulas have a somewhat pigeonhole principle-like flavour and one can intuitively argue that they would seem likely to be hard provided that every moderately large set of positive cardinality constraints involves variables from many different negative constraints.

This construction was further developed by Van Gelder and Spence in [VS10], where the variable partitioning is done in terms of an $n \times n$ matrix with 4 non-zero entries in each row and column except that one extra non-zero entry is added to some empty cell. The variables in the formula correspond to the non-zero entries, each row is a positive cardinality constraint on its non-zero entries just as before, and each column provides a negative cardinality constraint. Equivalently, this formula

[1] Provided that the solver does not reason in terms of cardinality constraints or systems of linear equations and does not introduce new variables to apply extended resolution, in which case the theoretical lower bound guarantees no longer apply.

[2] Tseitin formulas encode the principle that the sum of the vertex indegrees in an undirected graph is even.

can be constructed on a bipartite graph which is 4-regular on both sides except that one extra edge is added. In addition, there is a "no quadrangles" requirement in [VS10] that says that the graph contains no cycles of length 4. Just as above, it seems reasonable to believe that such formulas should be hard for resolution if the graph is a good expander. One such instance on 105 variables was issued by [VS10] as a "challenge formula" to be solved by any SAT solver in less than 24 hours, and in the concluding remarks the authors ask whether lower bounds on resolution length can be proven for formulas generated in this way.

Our Theoretical Results

We show that the formulas in [Spe10, VS10] are exponentially hard for resolution if the collection of constraints have a certain expansion property, and that random instances of these formulas are expanding in this sense with overwhelming probability. Let U denote the set of positive cardinality constraints and V the set of negative constraints. Then we can represent the formulas in [VS10] (and [Spe10]) as bipartite (multi-)graphs $G = (U \dot\cup V, E)$, where edges are identified with variables and $x = (u, v)$ is an edge in E if x occurs in both $u \in U$ and $v \in V$ (note that this is well-defined since each variable occurs in exactly one positive and one negative constraint). Informally, we obtain the following lower bound for resolution (see Theorem 7 for the formal statement).

Theorem 1 (Informal). *If a 4-regular bipartite (multi-)graph G with one extra edge added is a sufficiently good expander, then the formula in [VS10] generated from G (or in [Spe10] if G is a multigraph) requires resolution refutations of length $\exp\big(\Omega(n)\big)$. In particular, random instances of these formulas require resolution length $\exp\big(\Omega(n)\big)$ asymptotically almost surely.*

As a side note, we remark that the "no quadrangles" condition discussed above is not necessary (nor sufficient) for this theorem to hold—the more general notion of expansion is enough to guarantee that the formulas will be hard.

In one sentence, the proof works by reducing the formula to the pigeonhole principle on a 3-regular bipartite graph, which is then shown to be hard by a slight tweak of the techniques developed by Ben-Sasson and Wigderson [BW01]. A more detailed (if still incomplete) proof sketch is as follows. Start by fixing any complete matching in G (which can be shown to exist) and set all the matched edges plus the added extra edge to true. Also, set all remaining edges incident to the unique degree-5 vertex v^* on the right to false (this satisfies the negative constraint for v^*, which means that the corresponding clauses vanish). After this restriction, we are left with n constraints on the left which require that at least 1 out of the remaining 3 variables should be true, whereas on the right we have $n - 1$ constraints which all require that at most 1 remaining variable is true. But this is just a restricted PHP formula where each pigeon can go into one of three holes. Since we had a bipartite expander graph before restricting edges, and since not too many edges were removed,

the restricted graph is still an expander. Now we can argue along the lines of [BW01] to obtain a linear lower bound on the resolution width of refuting the formula, from which an exponential length lower bound follows (and since restrictions can only make formulas easier, this lower bound must also hold for the original formula).

In fact, using tools from [AR03] one can show that the formulas are hard not only for resolution but also for *polynomial calculus resolution* [ABRW02, CEI96], which adds the power of Gröbner basis computations to resolution.

Theorem 2 (Informal). *For* 4*-regular bipartite (multi-)graphs with one extra edge that are sufficiently good expanders the formulas in [Spe10, VS10] require refutations of size* $\exp\big(\Omega(n)\big)$ *in polynomial calculus resolution (PCR). In particular, randomly sampled instances of these formulas require PCR refutation size* $\exp\big(\Omega(n)\big)$ *asymptotically almost surely.*

The technical details of this argument get substantially more involved, however. Thus, although Theorem 1 is strictly subsumed by Theorem 2, we also present a self-contained proof of the former theorem since it is much cleaner and simpler.

Our Empirical Results

On the practical side, we report results from running some current state-of-the-art SAT solvers on random instances of the formulas constructed by Spence [Spe10] and Van Gelder and Spence [VS10], as well as on so-called *fixed bandwidth* versions of these formulas. The latter are formulas for which the non-zero entries on each row in the matrix appear on the diagonal and at some fixed (and small) horizontal offsets from the diagonal. Such matrices yield highly structured formulas, and as pointed out in [VS10] it is not hard to show that these formulas have refutations in polynomial length (and also constant width and space as defined in Section 2).

Our findings are that random instances of the formulas in [Spe10, VS10] are very hard, and become infeasible for slightly above 100 variables. As could be expected, the formulas in [VS10] are somewhat harder than the original formulas in [Spe10], since the former are guaranteed not to have any multi-edges in the bipartite graph representing the constraints and thus "spread out" variables better among different constraints. However, to our surprise the formulas that are hardest in practice are actually the ones generated from fixed bandwidth matrices. A priori, one possible explanation could be that although the formulas are theoretically easy, the constants hidden in the asymptotic notation are so bad that the instances are hard for all practical purposes. This appears not to be the case, however—when the SAT solver is explicitly given a good variable branching order the fixed bandwidth formulas are solved much more quickly. Thus, this raises the question whether this could perhaps be an example of formulas for which CDCL with current state-of-the-art heuristics fails to search effectively for resolution proofs. This stands in intriguing contrast to the theoretical results in [AFT11, PD11], which are usually interpreted as saying that CDCL essentially harnesses the full power of resolution.

We have also done limited experiments with feeding the formulas in [Spe10, VS10] to Sat4j [LP10], the latest version of which can detect (small) cardinality constraints [BLLM14]. It is not hard to see that if the SAT solver is given the power to count, then it could potentially figure out quickly that it cannot possibly be the case that a strict majority of the variables is both true and false simultaneously. Indeed, this is also what happens, and in particular Sat4j solves the challenge formula from [VS10] in less than a second.

Organization of This Paper

We start by reviewing some preliminaries in Section 2. In Section 3, we prove exponential lower bounds in resolution for the formulas by Van Gelder and Spence [VS10], and in Section 4 we describe how to modify this proof to also deal with Spence's original formulas [Spe10]. We extend all of these results to polynomial calculus resolution in Section 5. In Section 6, we report our experimental results. Section 7 contains some concluding remarks.

2 Proof Complexity Preliminaries

In what follows, we give a brief overview of the relevant proof complexity background. This material is standard and we refer to, e.g., the survey [Nor13] for more details. All formulas in this paper are in conjunctive normal form (CNF), i.e., consist of conjunctions of clauses, where a clause is a disjunction of positive literals (unnegated variables) and negative literals (negated variables, denoted by overline). It is convenient to view clauses as sets, so that there is no repetition of literals and order is irrelevant. A k-CNF formula has all clauses of size at most k, which is always assumed to be some fixed (and, in this paper, small) constant.

A *resolution refutation* $\pi : F \vdash \bot$ of a formula F (sometimes also referred to as a *resolution proof* for F) is an ordered sequence of clauses $\pi = (D_1, \ldots, D_\tau)$ such that $D_\tau = \bot$ is the empty clause without literals, and each line D_i, $1 \leq i \leq \tau$, is either one of the clauses in F (an *axiom* clause) or is derived from clauses D_j, D_k in π with $j, k < i$ by the *resolution rule*

$$\frac{B \vee x \qquad C \vee \overline{x}}{B \vee C} \tag{2.1}$$

(where the clause $B \vee C$ is the *resolvent* of the clauses $B \vee x$ and $C \vee \overline{x}$ on x). It is also sometimes technically convenient to add a *weakening rule*

$$\frac{B}{B \vee C} \tag{2.2}$$

that allows adding literals to a previously derived clause.

With every refutation π we can associate a graph G_π by having a sequence of vertices v_i labelled by the clauses D_i on a line in order of increasing i, and with edges from v_j and v_k to v_i (or from v_j to v_i) if D_i was derived by resolution from D_j

and D_k (or by weakening from D_j). Note that there might be several occurrences of a clause D in π, and if so each occurrence gets its own vertex in G_π. The *length* (or *size*) $L(\pi)$ of a resolution refutation π is the number of clauses in π counted with repetitions (i.e., the number of vertices in G_π). The *width* $W(C)$ of a clause C is the number of literals $|C|$, and the width $W(\pi)$ of a refutation π is the width of a largest clause in π. The *(clause) space* of a refutation at step i is the number of clauses C_j, $j < i$, with edges to clauses C_k, $k \geq i$, plus 1 for the clause C_i derived at this step. That is, intuitively space measures the number of clauses we need to keep in memory at step i, since they were derived before step i but will be used to infer new clauses after step i (or possibly at step i). The space $Sp(\pi)$ of a refutation is the maximum space over all steps in π. Taking the minimum over all resolution refutations of a formula F, we obtain the length $L_{\mathcal{R}}(F \vdash \bot)$, width $W_{\mathcal{R}}(F \vdash \bot)$, and space $Sp_{\mathcal{R}}(F \vdash \bot)$ of refuting F, respectively. It is not hard to show that all use of weakening can be eliminated from a resolution refutation without increasing any of these measures.

Resolution can be extended with algebraic reasoning corresponding to Gröbner basis computations to yield the proof system *polynomial calculus resolution (PCR)*, or more briefly just *polynomial calculus*.[3] For some fixed field $\mathbb{F}$ (which would be GF(2) in practical applications but can be any field in theory) we consider the polynomial ring $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$ with x and $\overline{x}$ as distinct formal variables,[4] and translate clauses $\bigvee_{x \in L^+} x \vee \bigvee_{y \in L^-} \overline{y}$ to monomials $\prod_{x \in L^+} x \cdot \prod_{y \in L^-} \overline{y}$. A *PCR refutation* π of F is then an ordered sequence of polynomials $\pi = (P_1, \ldots, P_\tau)$, expanded out as linear combinations of monomials, such that $P_\tau = 1$ and each line P_i, $1 \leq i \leq \tau$, is one of the following:

- a monomial encoding a clause in F;
- a *Boolean axiom* $x^2 - x$ or *complementarity axiom* $x + \overline{x} - 1$ for any variable x;
- a polynomial obtained from one or two previous polynomials in the sequence by applying a *linear combination*

$$\frac{Q \qquad R}{\alpha Q + \beta R} \tag{2.3}$$

  or *multiplication*

$$\frac{Q}{xQ} \tag{2.4}$$

  for any $\alpha, \beta \in \mathbb{F}$ and any variable x.

[3] Strictly speaking, PCR as defined in [ABRW02] is a slight generalization of polynomial calculus [CEI96], but in the current paper we will not be too careful in distinguishing between the two and the term "polynomial calculus" will refer to PCR unless specified otherwise.

[4] We remark that the distinct formal variables for negated literals, which is what [ABRW02] added on top of [CEI96], are there for theoretical reasons only in order to get a more well-behaved proof system. They would not appear in practical implementations of SAT solvers using Gröbner basis computations. On the theoretical side, they only make the proof system stronger, however, and so can only make our task of proving lower bounds harder.

Because of the Boolean axioms, we can assume without loss of generality that all polynomials in a PCR refutation are multilinear.

The *size* $S(\pi)$ of a PCR refutation π is the number of monomials in π (counted with repetitions), the *degree* $Deg(\pi)$ is the maximal degree of any monomial appearing in π, and *(monomial) space* $Sp(\pi)$ is defined in analogy with clause space, only counting monomials (with repetitions) instead of clauses. Taking the minimum over all PCR refutations of a CNF formula F, we define the size $S_{\mathcal{PCR}}(F \vdash \bot)$, degree $Deg_{\mathcal{PCR}}(F \vdash \bot)$, and space $Sp_{\mathcal{PCR}}(F \vdash \bot)$ of refuting F in PCR. When the proof system is clear from context, we will drop the subindices denoting resolution or PCR, respectively. It is straightforward to show that PCR can simulate resolution efficiently by simply mimicking the resolution steps in a refutation, and this simulation can be done without any noticeable blow up in size/length, degree/width, or space. There are formulas,[5] however, for which PCR can be exponentially stronger than resolution with respect to size/length.

A *restriction* ρ on F is a partial assignment to the variables of F. We use $\mathrm{Dom}(\rho)$ to denote the set of variables assigned by ρ. In a restricted formula $F\restriction_\rho$ (or refutation $\pi\restriction_\rho$) all clauses satisfied by ρ are removed and all other clauses have falsified literals removed. It is a well-known fact that restrictions preserve resolution refutations, so that if π is a resolution refutation of F, then $\pi\restriction_\rho$ is a refutation of $F\restriction_\rho$ (possibly using weakening) in at most the same length, width, and space. For polynomials, we think of 0 as true and 1 as false. Thus, if a restriction satisfies a literal in a monomial that monomial vanishes, and all falsified literals in a monomial get replaced by 1 and vanish. Again it holds that if π is a PCR refutation of F, then $\pi\restriction_\rho$ is a PCR refutation of $F\restriction_\rho$ (after a simple postprocessing step to take care of cancelling monomials and to adjust for that multiplication can only be done one variable at a time). This restricted refutation will have at most the same size, degree, and space (except possibly for a constant factor in size due to postprocessing multiplications).

3 Theoretically Hard Formulas on Expander Graphs

In this section, we present a lower bound for the formulas in [VS10]. Let us start by giving an explicit, formal definition of these formulas, which we will refer to as *subset cardinality formulas*.

Definition 3 (Subset cardinality formula). Suppose that $G = (U \dot\cup V, E)$ is a 4-regular bipartite (multi-)graph except that one extra edge has been added. Then the *subset cardinality formula* $SC(G)$ over G has variables $x_e, e \in E$, and clauses:

- $x_{e_1} \vee x_{e_2} \vee x_{e_3}$ for every triple e_1, e_2, e_3 of edges incident to $u \in U$,
- $\overline{x}_{e_1} \vee \overline{x}_{e_2} \vee \overline{x}_{e_3}$ for every triple e_1, e_2, e_3 of edges incident to $v \in V$.

[5] Examples of such formulas are Tseitin formulas and onto functional pigeonhole principles, but this is not relevant for the rest of this paper and so we do not discuss this in further detail here.

$$\begin{pmatrix}
1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1
\end{pmatrix}$$

(a) Matrix

$$\begin{aligned}
&(x_{1,1} \vee x_{1,2} \vee x_{1,4}) \\
\wedge\ &(x_{1,1} \vee x_{1,2} \vee x_{1,8}) \\
\wedge\ &(x_{1,1} \vee x_{1,2} \vee x_{1,10}) \\
\wedge\ &(x_{1,1} \vee x_{1,4} \vee x_{1,8}) \\
&\vdots \\
\wedge\ &(\overline{x}_{3,10} \vee \overline{x}_{7,10} \vee \overline{x}_{9,10}) \\
\wedge\ &(\overline{x}_{3,10} \vee \overline{x}_{7,10} \vee \overline{x}_{10,10}) \\
\wedge\ &(\overline{x}_{3,10} \vee \overline{x}_{9,10} \vee \overline{x}_{10,10}) \\
\wedge\ &(\overline{x}_{7,10} \vee \overline{x}_{9,10} \vee \overline{x}_{10,10})
\end{aligned}$$

(b) Formula

Figure 2: Example of a fixedbandwidth matrix of size 10 and the corresponding formula.

An example of a formula based on the fixed bandwidth matrix is given in Figure 2. As noted before, an easy counting argument shows that these formulas are unsatisfiable. Intuitively, the hardness of proving this unsatisfiability should depend on the structure of the underlying graph G. We remind the reader that compared to [VS10], the "no quadrangles" condition mentioned in Section 1 is missing in Definition 3. This is because this condition is neither necessary nor sufficient to obtain lower bounds. Expressed in terms of the graph G, what quadrangle-freeness means is that there are no 4-cycles, which is essentially saying that no constraints in G have a very "localized structure." However, the fixed bandwidth formulas already discussed in Section 1 can be constructed to be quadrangle-free, but are still guaranteed to be easy for resolution. Therefore, in order for our lower bound proof to go through we need the more general condition that the graph G should be an expander as defined next.

Definition 4 (Expander). A bipartite graph $G = (U \dot\cup V, E)$ is an (s, δ)-*expander* if for each vertex set $U' \subseteq U, |U'| \leq s$, it holds that $|N(U')| \geq \delta|U'|$, where $N(U') = \{v \in V \mid \exists (u, v) \in E \text{ for } u \in U'\}$ is the set of neighbours of U'.

The key idea in our lower bound proof is to apply a suitably chosen restriction to reduce subset cardinality formulas to so-called *graph pigeonhole principle formulas* $PHP(G)$. These formulas are also defined in terms of bipartite graphs $G = (U \dot\cup V, E)$ and encode that every "pigeon" vertex on the left, i.e., in U, needs to have at least one of the edges incident to it set to true, while every "hole" vertex on the right, i.e., in V, must have at most one edge incident to it set to true. Formally, $PHP(G)$

is a CNF formula over variables $x_{u,v}$, for every edge in $(u, v) \in E$, consisting of the following axioms:

$$\bigvee_{v \in N(u)} x_{u,v} \quad \text{for all } u \in U \qquad \text{(pigeon axioms)} \tag{3.1}$$

$$\overline{x}_{u,v} \vee \overline{x}_{u',v} \quad \text{for all } v \in V \text{ and } (u, v), (u', v) \in E, \text{ where } u \neq u' \qquad \text{(hole axioms)} \tag{3.2}$$

Ben-Sasson and Wigderson [BW01] showed that random instances of such formulas are hard for resolution if the left degree is at least 5, and modifying their techniques slightly we prove that left degree 3 is sufficient provided that the graphs have good enough expansion. The proof is by showing a resolution width lower bound and then applying the lower bound on length in terms of width in [BW01]. An analogous result can also be proven for polynomial calculus by using techniques from Alekhnovich and Razborov [AR03] to obtain a degree lower bound and then applying the lower bound on size in terms of degree in Impagliazzo et al. [IPS99], which yields the following lemma.

Lemma 5. *Suppose that $G = (U \dot\cup V, E)$ is a 3-regular $\left(\epsilon n, \frac{3}{2} + \delta\right)$-expander for some constant $\epsilon, \delta > 0$ and $|U| = |V| = n$, and let G' be the graph obtained by removing any vertex from V in G and its incident edges. Then the resolution refutation length of the graph pigeonhole principle $PHP(G')$ is $\exp(\Omega(n))$, and the same bound holds for PCR size.*

We first show how Lemma 5 can be used to establish the lower bound for subset cardinality formulas. In order to do this, we need the following standard lemma for regular bipartite graphs, the proof of which is provided for completeness.

Lemma 6 ([BM08]). *Every d-regular bipartite graph has a perfect matching.*

Proof. Let $U' \subseteq U$ be an arbitrary set of left vertices and let E_1 be the set of edges incident to U'. Let $V' = N(U')$ and let E_2 be the set of edges incident to V'. It holds that $E_2 \supseteq E_1$ and, hence, $|E_2| \geq |E_1|$. By the d-regularity of the graph we have that $|E_1| = d|U'|$ and $|E_2| = d|V'|$. Hence, it holds that $|V'| \geq |U'|$ for every $U' \subseteq U$, which by Hall's theorem implies the existence of a matching. □

Theorem 7. *Suppose that $G = (U \dot\cup V, E)$ is a 4-regular $\left(\epsilon n, \frac{5}{2} + \delta\right)$-expander for $|U| = |V| = n$ and some constants $\epsilon, \delta > 0$, and let G' be obtained from G by adding an arbitrary edge from U to V. Then any polynomial calculus refutation of $SC(G')$ must have size $\exp\big(\Omega(n)\big)$ (and hence the same lower bound holds for resolution length).*

Proof. We want to restrict the subset cardinality formula $SC(G')$ to get a graph pigeonhole principle formula. By Lemma 6 it holds that G has a perfect matching.

Fix such a matching M and let $M' = M \cup \{(u', v')\}$, where (u', v') denotes the edge added to G. We apply the following restriction ρ to $SC(G')$:

$$\rho\left(x_{(u,v)}\right) = \begin{cases} \top & \text{if } (u,v) \in M', \\ \bot & \text{if } v = v' \text{ and } (u,v') \notin M', \\ * & \text{otherwise (i.e., the variable is unassigned).} \end{cases} \tag{3.3}$$

This reduces the original formula $SC(G')$ to $PHP(G'')$ on the graph G'' obtained by removing the matching M and also the vertex v' with incident edges from G. To see this, consider what happens with the clauses encoding the constraints.

For every vertex $u \in U \setminus \{u'\}$, which has four edges $e_i, 1 \leq i \leq 4$, incident to it, we have the clauses

$$\{x_{e_1} \vee x_{e_2} \vee x_{e_3}, x_{e_1} \vee x_{e_2} \vee x_{e_4}, x_{e_1} \vee x_{e_3} \vee x_{e_4}, x_{e_2} \vee x_{e_3} \vee x_{e_4}\} \tag{3.4}$$

in $SC(G')$. After applying the restriction ρ, the one edge that is in the matching M will be set to true, satisfying all of these clauses but one. For instance, if $e_4 \in M$ then only the clause $x_{e_1} \vee x_{e_2} \vee x_{e_3}$ remains, which corresponds to the pigeon axiom for the vertex u in G''. If in addition u is one of the vertices neighbouring v', the remaining constraint will shrink to a 2-clause. The constraint corresponding to u' is similarly reduced. In this case, we have five incident edges $e_i, 1 \leq i \leq 5$, and two of them are set to true. If, for instance, we have $e_4 \in M$ and $e_5 = (u', v')$, then the only clause that is not satisfied is $x_{e_1} \vee x_{e_2} \vee x_{e_3}$, which corresponds to the pigeon axiom for the vertex u' in G''.

For a constraint $v \in V \setminus \{v'\}$ with neighbours $e_i, 1 \leq i \leq 4$, the clause set is the same as for $U \setminus \{u'\}$ except that every variable is negated. If $e_4 \in M$, then after the restriction we are left with the set of clauses

$$\{\overline{x}_{e_1} \vee \overline{x}_{e_2} \vee \overline{x}_{e_3}, \overline{x}_{e_1} \vee \overline{x}_{e_2}, \overline{x}_{e_1} \vee \overline{x}_{e_3}, \overline{x}_{e_2} \vee \overline{x}_{e_3}\} \ . \tag{3.5}$$

where the last three clauses are the hole axioms for the vertex v in G'' and the first clause can be ignored since it follows by weakening of any of the other clauses. Since ρ satisfies the constraint v' the clauses encoding this constraint vanish. This shows that $SC(G')\restriction_\rho$ is indeed equal to $PHP(G'')$.

Now all that remains is to observe that G'' can be obtained by removing a right vertex from a 3-regular bipartite $(\epsilon n, \frac{3}{2} + \delta)$-expander. This is so since deleting the matching M from G decreases all vertex degrees from 4 to 3 and lowers the expansion factor by at most an additive 1. Applying Lemma 5 we conclude that $PHP(G'')$ requires polynomial calculus size (and hence resolution length) $\exp\big(\Omega(n)\big)$. As restrictions do not increase the length/size of refutations, the same lower bound must hold also for $SC(G')$. □

It remains to prove Lemma 5. Below we give a full proof of the lemma for resolution, while the argument for polynomial calculus is given in Section 5. For both resolution and polynomial calculus we need a stronger notion of expansion as defined next.

Definition 8 (Boundary expander). A bipartite graph $G = (U \dot\cup V, E)$ is an (s, δ)-*boundary expander* if for every set of vertices $U' \subseteq U, |U'| \leq s$, it holds that $|\partial(U')| \geq \delta|U'|$, where $v \in \partial(U')$ if there is exactly one vertex $u \in U'$ that is a neighbour of v.

We have the following connection between usual expansion and boundary expansion, a proof of which is provided for completeness.

Proposition 9. *Every d-regular (s, δ)-expander is also an $(s, 2\delta - d)$-boundary expander.*

Proof. For any set $U' \subseteq U, |U'| \leq s$, we have that $d|U'|$ edges are spread among at least $\delta|U'|$ neighbours. After each of the neighbours gets one edge we have at most $(d - \delta)|U'|$ edges left to spread. Hence, we are guaranteed that at least $\delta|U'| - (d-\delta)|U'| = (2\delta - d)|U'|$ neighbours do not get a new edge and are neighbours of exactly one vertex in U'. □

With this stronger notion of expansion and the following theorem from [BW01] we can prove Lemma 5 by lower bounding refutation width.

Theorem 10 ([BW01]). *For any constant k and an unsatisfiable k-CNF formula F with n variables it holds that $L(F \vdash \bot) = \exp\big(\Omega\big(W(F \vdash \bot)^2/n\big)\big)$.*

Proof of Lemma 5 for resolution. Since by Proposition 9 G is an $(\epsilon n, 2\delta)$-boundary expander, even after removing a vertex in V it must hold for G' that every set of vertices $U'' \subseteq U$, $|U''| \leq \epsilon n$ satisfies $|\partial_{G'}(U'')| \geq 2\delta|U''| - 1$.

Let us also observe that the connected component $G^c = (U^c \dot\cup V^c, E^c)$ of G to which the vertex v' belongs must be a 3-regular graph with $|U^c| > \epsilon n$. This is so since if $|U^c| \leq \epsilon n$, it would follow from the expansion of G that $|V^c| = |N_{G^c}(U^c)| \geq \big(\frac{3}{2} + \delta\big)|U^c| > |U^c|$. But $|U^c| \neq |V^c|$ implies that G^c cannot be a 3-regular bipartite graph, which is a contradiction. Furthermore, for every proper subset $U'' \subsetneq U^c$ it must hold that $|N(U'')| > |U''|$, since otherwise U'' and its neighbours $N(U'')$ would form a disconnected component in G^c. Hence, when we remove the vertex v' from G^c we have $|N(U'')| \geq |U''|$ for every proper subset $U'' \subsetneq U^c$. By Hall's theorem, this implies that every proper subset $U'' \subsetneq U^c$ has a matching in G^c. This shows that any refutation of $PHP(G')$ must use all the pigeons in G^c, i.e., at least ϵn pigeon axiom clauses, to show that $PHP(G')$ is unsatisfiable, since the formula becomes satisfiable if just one of these pigeon axioms is removed.

Now we can employ the progress measure on refutations developed in [BW01] to show that the width of refuting $PHP(G')$ is lower bounded by $\epsilon\delta n - 1$. To every clause C_i in a refutation we assign a measure that represents the size of a minimal subset U'' of pigeons U' such that the formula $PHP(G'{\restriction_{U''}})$ implies C_i, where $PHP(G'{\restriction_{U''}})$ is a subformula of $PHP(G')$ that consists of pigeon axioms for vertices in U'' and all hole axioms.

Axioms have measure at most 1, while the measure of an empty clause is greater than ϵn, as every subformula of $PHP(G')$ that has at most ϵn pigeon axioms is

satisfiable. Moreover, by a simple application of the union bound it is easy to see that the progress measure can at most double at each resolution step. Hence, in any refutation of $PHP(G')$ there is a clause C^* having measure between $\epsilon n/2$ and ϵn.

We can now argue by expansion and show that C^* has at least $\epsilon\delta n - 1$ variables. Let U^* be a set of pigeons that defines the measure of C^*, that is the measure of C^* is equal to $|U^*|$ and $PHP(G'\restriction_{U^*})$ implies C^*. Fix v^* to be some hole in $\partial_{G'}(U^*)$ and let u^* be the unique pigeon in U^* that has v^* as one of its neighbors. By minimality of U^*, there exists an assignment that falsifies C^*, but satisfies the subformula $PHP(G'\restriction_{U^*\setminus\{u^*\}})$ where we removed the pigeon axiom for u^*. If we modify this assignment by setting x_{u^*,v^*} to true and all other x_{u,v^*} that mention hole v^* to false, we get an assignment that satisfies $PHP(G'\restriction_{U^*})$. Hence, this assignment must also satisfy C^*. As we changed only the variables that mention v^*, it follows that C^* needs to contain at least one of the variables x_{u,v^*} for the hole $v^* \in \partial_{G'}(U^*)$. This holds for every hole in $\partial_{G'}(U^*)$ and hence C^* has at least $2\delta|U^*| - 1$ variables, which is greater than $\epsilon\delta n - 1$.

By appealing to the lower bound on length in terms of width in Theorem 10 we obtain a lower bound on the resolution refutation length of $\exp\big(\Omega(n)\big)$. □

This proves that the formulas in [VS10] are hard for resolution if the underlying graph is an expander. In order to establish that randomly sampled instances of such formulas are hard, we just need the fact that randomly sampled graphs are likely to be expanders. This follows by a theorem from [HLW06] in which it is proved that random regular bipartite graphs are excellent expanders almost surely, except that in their model these graphs are not necessarily simple but can have multiple edges. However, if one conditions on the fact that the produced graph is simple, then the resulting distribution is uniform over random graphs. Since it can also be shown for a graph sampled randomly according to this distribution that the probability that the graph is simple is bounded away from zero (see, e.g., [BS13, Jan13]), we obtain the result that we need that a graph sampled uniformly at random from the set of all 4-regular bipartite graphs is an expander almost surely.

Theorem 11 ([HLW06]). *Let $d \geq 3$ be a fixed integer. Then for every $\delta, 0 < \delta < \frac{1}{2}$, there exists an $\epsilon > 0$ such that almost all d-regular bipartite graphs G with n vertices on each side are $\big(\epsilon n, d - \frac{3}{2} + \delta\big)$-expanders.*

Corollary 12. *The formula $SC(G)$ for a random 4-regular bipartite graph G with an arbitrary extra edge added requires polynomial calculus refutations (and hence also resolution refutations) of exponential size asymptotically almost surely.*

Proof. Use Theorem 11 with $d = 4$ together with Theorem 7. □

In the next section we discuss how to extend these results to formulas from [Spe10], which are defined on random permutations instead of bipartite graphs.

4 Theoretically Hard Formulas on Random Permutations

Previously we have defined subset cardinality formulas for bipartite graphs. However, in order to define formulas in [Spe10] we need to extend the previous definition to a definition based on permutations. To achieve this we define the following relation between permutations and bipartite multigraphs, where we use $[n]$ to denote the set $\{1, 2, \ldots, n\}$.

Definition 13 (Multigraph from permutation). For a permutation σ on $[4n]$, $G(\sigma) = (U \dot\cup V, E)$ is a bipartite multigraph such that

- $U = \{\{\sigma(1), \sigma(2), \sigma(3), \sigma(4)\}, \{\sigma(5), \sigma(6), \sigma(7), \sigma(8)\}, \ldots, \{\sigma(4n-3), \sigma(4n-2), \sigma(4n-1), \sigma(4n)\}\}$,
- $V = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \ldots, \{4n-3, 4n-2, 4n-1, 4n\}\}$, and
- for every $u \in U$ and $v \in V$, there are $|u \cap v|$ edges (u, v) in E.

If σ is a permutation on $[4n+1]$, then the last sets of U and V additionally have elements $\sigma(4n+1)$ and $4n+1$, respectively. That is, the last elements are $\{\sigma(4n-3), \sigma(4n-2), \sigma(4n-1), \sigma(4n), \sigma(4n+1)\} \in U$ and $\{4n-3, 4n-2, 4n-1, 4n, 4n+1\} \in V$. The edges are defined accordingly.

We can view the multigraph from the previous definition as assigning 4 outgoing edges to every vertex on the left and right, except for the last vertices which might have 5 edges assigned to them. To define how the edges connect to each other we number the edges on the right from 1 to $4n+1$, while on the left we number them according to the permutation σ. The two edges are then merged into a single edge if they share the same number.

In Definition 13 we defined the multigraph for permutations on $[4n]$ and on $[4n+1]$ as we use both in the proof, although only the $4n+1$ case gives us the formulas in [Spe10].

Definition 14 ([Spe10]). For a permutation σ on $[4n+1]$ numbers, the formula $SC^*(\sigma)$ is a subset cardinality formula on the multigraph $G(\sigma)$.

Let us now translate the definitions and proofs of the previous section from the graph case to the multigraph case. Ordinary and boundary expansion for multigraphs are defined similarly as in the case of graphs, and the relation between the two is the same as in Proposition 9.

Definition 15 (Expansion). A bipartite multigraph $G = (U \dot\cup V, E)$ is an (s, δ)-expander if for every set of vertices $U' \subseteq U, |U'| \leq s$, it holds that $|N(U')| \geq \delta|U'|$.

Definition 16 (Boundary expansion). A bipartite multigraph $G = (U \dot\cup V, E)$ is an (s, δ)-boundary expander if for every set of vertices $U' \subseteq U, |U'| \leq s$, it holds that $|\partial(U')| \geq \delta|U'|$, where $v \in \partial(U')$ if there is exactly one vertex $u \in U'$ that is the neighbor of v (possibly connected to v by multiple edges).

Proposition 17. *Every d-regular multigraph (s, δ)-expander is also a multigraph $(s, 2\delta - d)$-boundary expander.*

Proof. For any set $U' \subseteq U, |U'| \leq s$, we have that $d|U'|$ edges are spread among at least $\delta|U'|$ neighbors. After each of the neighbors gets one edge we have at most $(d - \delta)|U'|$ edges left to spread. Hence, we are guaranteed that at least $\delta|U'| - \big((d - \delta)|U'|\big) = (2\delta - d)|U'|$ neighbors do not get a new edge and are neighbors of exactly one vertex in U'. □

Furthermore it is easy to check that the proof of Lemma 6, which states that we can always find a matching in a regular bipartite graph, works for multigraphs.

Lemma 18 ([BM08]). *Every regular bipartite multigraph has a perfect matching.*

Because we are dealing with multigraphs, restricting $SC^*(\sigma)$ might not yield a pigeonhole principle formula that is based on a 3-regular graph. Nevertheless, the regularity requirement in Lemma 5 is not essential and can be replaced. As regularity was only used to establish the satisfiability of every subformula of bounded size, we can exchange it with the requirement that every subset U' of pigeons of bounded size has a matching. With this modification to Lemma 5 we have the same lower bound.

Lemma 19. *Suppose that $G = (U \dot\cup V, E)$ is a bipartite graph with bounded left degree and that $\epsilon, \delta > 0$ are constants such that*

- *$|U| = n$ and $|V| = n - 1$,*
- *for every set $U' \subseteq U$ of size $|U'| \leq \epsilon n$, there is a matching of U' into V, and*
- *for every set $U' \subseteq U$ of size $|U'| \leq \epsilon n$, it holds that $|\partial(U')| \geq 2\delta|U'| - 1$.*

Then, every resolution/polynomial calculus resolution refutation of the graph pigeonhole principle PHP(G) has length/size $\exp(\Omega(n))$.

Before we can finally state our theorem, we need one more definition. This one provides a mapping from permutations on $[n]$ to permutations on $[n - 1]$.

Definition 20. For a permutation σ on $[n]$, a permutation reduction R_σ is a function that returns a permutation σ' on $[n - 1]$ defined as follows:

$$\sigma'(i) = \begin{cases} \sigma(n), & \sigma(i) = n \\ \sigma(i), & \text{otherwise} \end{cases},$$

for every i in $[n - 1]$.

Thus, permutation σ' is formed by taking σ and replacing the occurrence of n in σ with its last element, that is $\sigma(n)$. Now we can state our lower bound for formulas in [Spe10].

Theorem 21. *Suppose that σ is a permutation on $[4n+1]$ such that $G(R_\sigma(\sigma))$ is a multigraph $\left(\epsilon n, \frac{5}{2}+\delta\right)$-expander for some $\epsilon, \delta > 0$. Then any polynomial calculus refutation of $SC^*(\sigma)$ must have size $\exp(\Omega(n))$ (and hence the same lower bound holds for resolution length).*

Proof. As in the case of $SC(G)$, we restrict $SC^*(\sigma)$ to a graph pigeonhole principle formula. In order for us to be able to uniquely refer to the edges of the multigraph $G(\sigma)$ we label each edge with an index i, so that an edge e_i connects vertices $u \in U$ and $v \in V$ only if i is in both u and v.

Let us now relate the multigraph $G(R_\sigma(\sigma))$ to $G(\sigma)$. If $\sigma(4n+1) = 4n+1$, then $G(R_\sigma(\sigma))$ is the multigraph we get from $G(\sigma)$ by removing the edge e_{4n+1}. Otherwise, $G(R_\sigma(\sigma))$ is the multigraph we get from $G(\sigma)$ by removing two distinct edges $e_{\sigma(4n+1)}$ and e_{4n+1}, and then connecting the degree 3 vertices that result from this removal.

For the multigraph $G(\sigma)$, let u^* and v^* be degree 5 vertices in U and V, respectively. Let v_{u^*} be the vertex in V that contains $\sigma(4n+1)$, so that v_{u^*} is connected to u^* by the edge $e_{\sigma(4n+1)}$. Note that it is possible that v_{u^*} is equal to v^*. We now find a matching M in $G(\sigma)$ that does not contain neither of the edges $e_{\sigma(4n+1)}$ nor e_{4n+1}, which might be the same edge. This follows by applying Lemma 18 to the graph $G(R_\sigma(\sigma))$ and noting that every edge in $G(R_\sigma(\sigma))$, except at most one, appears also in $G(\sigma)$. In forming M we can avoid this added edge by applying the lemma twice in succession and, out of the two produced matchings, picking the matching that does not contain the edge which is not in $G(\sigma)$. Let M' then be equal to $M \cup \{e_{\sigma(4n+1)}\}$.

We apply the following restriction ρ to the formula $SC^*(\sigma)$:

$$\rho(x_e) = \begin{cases} \top & \text{if } e \in M' \\ \bot & \text{if } e \text{ is incident to } v_{u^*} \text{ and } e \notin M' \\ * & \text{otherwise} \end{cases} . \tag{4.1}$$

This restriction reduces the original formula to the graph pigeonhole principle formula on the multigraph expander G'. The analysis of what happens to clauses is similar to the proof of Theorem 7. Every vertex in U needs at least 2 of its edges to be true, except u^* which needs at least 3. After setting all the edges in M' to true, the vertices in U require just 1 of their remaining edges to be true. Hence, they are equal to pigeon axioms. The vertices in V initially need at most 2 of their edges to be true and setting the edges in the matching M to true drops this bound to at most 1. Furthermore, setting $e_{\sigma(4n+1)}$ to true ensures that the vertex v_{u^*}, incident to $e_{\sigma(4n+1)}$, can only be satisfied by setting its remaining edges to false. Hence, in the restriction ρ we satisfy v_{u^*}, while the remaining vertices in V are left with constraints that correspond to the hole axioms.

To prove that the multigraph G' is a good expander, we show that it has as a subgraph a multigraph G'' obtained by removing the matching M and the vertex v_{u^*} from the multigraph $G(R_\sigma(\sigma))$. The restriction ρ removes the edges in

the matching M and the vertex v_{u^*} from the multigraph $G(\sigma)$. The only edge that might exist in $G(R_\sigma(\sigma))$ and not in $G(\sigma)$ is the edge that we add to connect the degree 3 vertices that result from removing e_{4n+1} and $e_{\sigma(4n+1)}$. But, this added edge is incident to v_{u^*} and hence we remove it when producing G''. Also, the matching M exists in $G(R_\sigma(\sigma))$ as well, so it gets removed to get G''. Hence, G'' is a subgraph of G'. To simplify the rest of the argument we assume that G' is actually equal to G''. Note that the additional edge G' could have can only help out with expansion and the existence of the matching.

As $G(R_\sigma(\sigma))$ is a 4-regular multigraph $(\epsilon n, \frac{5}{2} + \delta)$-expander, by Proposition 17 it holds that every subset of vertices $U'' \subseteq U', |U''| \leq \epsilon n$, in G' has a boundary of size $|\partial(U'')| \geq 2\delta|U''| - 1$. Also, by the argument analogous to the one in Lemma 5 we have that every subset $U'' \subseteq U', |U''| \leq \epsilon n$, in G' has a matching. Thus, the conditions of Lemma 19 are almost satisfied, except that G' is a multigraph and not a graph. The last issue can be solved by applying another restriction that for each pair of vertices $u \in U'$ and $v \in V'$ sets to false all of the edges between u and v except one. The resulting formula is a graph pigeonhole principle formula that satisfies conditions in Lemma 19.

Hence, by applying Lemma 19 we conclude that any polynomial calculus refutation of $SC^*(\sigma)$, where $G(R_\sigma(\sigma))$ is an expander, requires size $\exp(\Omega(n))$. □

To prove that randomly generated $SC^*(\sigma)$ formulas require exponential size to refute note that the proof of Theorem 11 in [HLW06] already works for multigraphs $G(\sigma)$, where σ is a random permutation on $[4n]$. Hence, the only difference is that $SC^*(\sigma)$ formulas are based on random permutations on $[4n + 1]$ instead of $[4n]$, but the following relation between σ and $R_\sigma(\sigma)$ resolves this issue.

Proposition 22. *If σ is chosen uniformly at random from the set of all permutations on $[n]$, then $R_\sigma(\sigma)$ is uniformly distributed among the set of all permutations on $[n-1]$.*

Proof. We show that for any permutation σ' on $[n-1]$, there are n distinct permutations σ_{i^*} on $[n]$ such that $R_\sigma(\sigma_{i^*}) = \sigma'$. To see this, set i^* to be any index between 1 and n, and construct the permutation σ_{i^*} on n as follows

$$\sigma_{i^*}(i) = \begin{cases} n & \text{if } i = i^* \\ \sigma'(i^*) & \text{if } i = n \\ \sigma'(i) & \text{otherwise} \end{cases} . \tag{4.2}$$

It is easy to see that $R_\sigma(\sigma_{i^*}) = \sigma'$ for every i^*, and that all σ_{i^*} are distinct. Moreover, the set of permutations σ_{i^*} includes all the permutations on $[n]$ that can map to σ'. This follows by counting. As there are n distinct permutations σ_{i^*} for every permutation σ' on $[n-1]$, taking all of them gives us $n!$ distinct permutations accounting for all permutations on $[n]$.

Hence, for a uniformly random distribution over permutations σ on $[n]$, the reduction $R_\sigma(\sigma)$ produces a uniformly random distribution over permutations on $[n-1]$. □

Corollary 23. *The formula $SC^*(\sigma)$ for a random permutation on $[4n+1]$ requires polynomial calculus refutations of exponential size asymptotically almost surely.*

Proof. As previously noted, the proof of Theorem 11 works for multigraphs constructed from random permutations on $[4n]$. Also, by Proposition 22 it follows that for every uniformly random permutation σ on $[4n+1]$, the permutation $R_\sigma(\sigma)$ is uniformly random on $[4n]$ and, hence, the multigraph $G(R_\sigma(\sigma))$ is an $\left(\epsilon n, \frac{5}{2}+\delta\right)$-expander asymptotically almost surely. Therefore, by Theorem 21, the formula $SC^*(\sigma)$ based on a random permutation σ requires $\exp(\Omega(n))$ size to refute asymptotically almost surely. □

5 Size Lower Bound for Polynomial Calculus

In order to prove the lower bound from Lemma 19, we only need to slightly modify the lower bound for degree from [AR03] and the result immediately follows. However, in subsequent work [MN15] we have developed a generalized approach to proving degree lower bounds, which we use to prove the lower bound in this work. To transform the degree lower bound into the size lower bound we will use the following theorem.

Theorem 24 ([IPS99]). *Let F be an unsatisfiable CNF formula of width $W(F)$ over n variables. Then*

$$S_{\mathcal{PCR}}(F \vdash \bot) = \exp\left(\Omega\left(\frac{(Deg_{\mathcal{PCR}}(F \vdash \bot) - W(F))^2}{n}\right)\right) .$$

The key idea in [MN15] is to construct a bipartite graph that makes explicit the constraints encoded in a CNF formula. In doing this we need to keep track of how certain partial assignments affect the clauses of the formula. We use the notation $Vars(C)$ and $Vars(F)$ to denote the set of variables appearing in a clause C or a formula F, respectively.

Definition 25. We say that a partial assignment ρ *respects* a CNF formula E if for every clause C in E either $Vars(C) \cap \mathrm{Dom}(\rho) = \emptyset$ or ρ satisfies C. A set of variables V respects a CNF formula E if there exists an assignment ρ with domain $\mathrm{Dom}(\rho) = V$ that respects E.

Definition 26 (Respectful satisfaction). Let F and E be CNF formulas and let V be a set of variables. We say that F is *E-respectfully satisfiable by V* if there exists an assignment ρ with domain $\mathrm{Dom}(\rho) = V$ that satisfies F and respects E, and that such an assignment ρ *E-respectfully satisfies F*.

Another way of stating the previous definition is that we have an autarky ρ for E (i.e., an assignment which satisfies all clauses in E which it touches) that satisfies the formula F.

Definition 27 (Bipartite $(\mathcal{U},\mathcal{V})_E$-graph [MN15]). Let E be a CNF formula, $\mathcal{U}$ be a set of CNF formulas, and $\mathcal{V}$ be a family of sets of variables V that respect E. Then the *(bipartite) $(\mathcal{U},\mathcal{V})_E$-graph* is a bipartite graph with left vertices $F \in \mathcal{U}$, right vertices $V \in \mathcal{V}$, and edges between F and V if $Vars(F) \cap V \neq \emptyset$.

Furthermore, for every edge (F,V) in the graph we say that F and V are *E-respectful neighbours* if F is E-respectfully satisfiable by V. Otherwise, they are *E-disrespectful neighbours*.

To denote the set of all neighbours $V \in \mathcal{V}$ of a formula F in the $(\mathcal{U},\mathcal{V})_E$-graph, we use the standard graph notation $N(F)$.

Definition 28 (Respectful boundary). For a $(\mathcal{U},\mathcal{V})_E$-graph and a subset $\mathcal{U}' \subseteq \mathcal{U}$, the *$E$-respectful boundary* $\partial_E(\mathcal{U}')$ of $\mathcal{U}'$ is the family of variable sets $V \in \mathcal{V}$ such that each $V \in \partial_E(\mathcal{U}')$ is an E-respectful neighbour of some clause set $F \in \mathcal{U}'$ but is not a neighbour (respectful or disrespectful) of any other clause set $F' \in \mathcal{U}' \setminus \{F\}$.

As it makes the notation more convenient, we will interpret subsets $\mathcal{U}' \subseteq \mathcal{U}$ as CNF formulas $\bigwedge_{F\in\mathcal{U}'} \bigwedge_{C\in F} C$. The lower bound in [MN15] follows if the $(\mathcal{U},\mathcal{V})_E$-graph is a good expander and if every variable does not appear in too many sets in $\mathcal{V}$, as defined next.

Definition 29 (Respectful boundary expander). A $(\mathcal{U},\mathcal{V})_E$-graph is said to be an *(s,δ,ξ,E)-respectful boundary expander*, or just an *(s,δ,ξ,E)-expander* for brevity, if for every set $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$, it holds that $|\partial_E(\mathcal{U}')| \geq \delta|\mathcal{U}'| - \xi$.

Definition 30. The *overlap* of a variable x with respect to a family of variable sets $\mathcal{V}$ is $ol(x,\mathcal{V}) = |\{V \in \mathcal{V} : x \in V\}|$ and the overlap of $\mathcal{V}$ is $ol(\mathcal{V}) = \max_x\{ol(x,\mathcal{V})\}$, i.e., the maximum number of sets $V \in \mathcal{V}$ containing any particular variable x.

The concept above is also referred to as the "maximum degree" in the literature.

Theorem 31 ([MN15]). *Let a $(\mathcal{U},\mathcal{V})_E$-graph be an (s,δ,ξ,E)-expander with overlap $ol(\mathcal{V}) = d$ and such that for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$, it holds that $\mathcal{U}' \wedge E$ is satisfiable. Then any polynomial calculus refutation of the formula $\mathcal{U} \wedge E$ requires degree strictly greater than $(\delta s - 2\xi)/(2d)$.*

To prove that the pigeonhole principle formula from Lemma 19 gives a lower bound on polynomial calculus degree, we just need to construct the appropriate $(\mathcal{U},\mathcal{V})_E$-graph and prove that it satisfies the conditions of Theorem 31.

Proof of Lemma 19. The $(\mathcal{U},\mathcal{V})_E$-graph for $PHP(G)$ is formed by taking $\mathcal{U}$ to be the set of pigeon axioms (3.1), E to consist of the hole axioms (3.2), and $\mathcal{V}$ to be

the collection of variable sets $V_v = \{x_{u,v} \mid u \in N(v)\}$ partitioned with respect to the holes $v \in V$.

We can check that this $(\mathcal{U}, \mathcal{V})_E$-graph is isomorphic to the graph G and that all neighbours in the $(\mathcal{U}, \mathcal{V})_E$-graph are E-respectful. First note that the variables in a pigeon axiom $F_u \in \mathcal{U}$ mention exactly the holes $N(u)$ from the neighbourhood of $u \in U$ in G. Hence, it holds that $N(F_u) = \{V_v \in \mathcal{V} \mid v \in N(u)\}$. To see that all neighbours are E-respectful take a pigeon axiom $F_u \in \mathcal{U}$ and one of its neighbouring variables sets $V_v \in N(F_u)$. Then, the assignment ρ that sets $x_{u,v}$ to true and all other variables in V_v to false E-respectfully satisfies F_u. The pigeon axiom F_u is satisfied because it contains the variable $x_{u,v}$ that is set to true, while every hole axiom in E that mentions $x_{u,v}$ contains another variable $x_{u',v} \in V_v$ which is set to false and hence the hole axioms are satisfied as well. Moreover, this argument shows that all $V_v \in \mathcal{V}$ are E-respectful. Thus, we have shown that the $(\mathcal{U}, \mathcal{V})_E$-graph is isomorphic to G and that all neighbours are E-respectful.

It follows that the $(\mathcal{U}, \mathcal{V})_E$-graph is an $(\epsilon n, 2\delta, 1, E)$-respectful boundary expander by the assumption of the theorem. To apply Theorem 31 we are left with showing that $\mathcal{U}' \wedge E$ is satisfiable for every $\mathcal{U}' \subseteq \mathcal{U}, |\mathcal{U}'| \leq \epsilon n$. To establish this claim we use the assumption of the lemma which states that any set U' of size $|U'| \leq \epsilon n$ has a matching in G. For any $\mathcal{U}', |\mathcal{U}'| \leq \epsilon n$, we have a corresponding set of pigeons U' in G such that $\mathcal{U}' = \{F_u \mid u \in U'\}$ of the same size $|U'| = |\mathcal{U}'|$. We know that this set has some matching M in G. We can satisfy $\mathcal{U}' \wedge E$ by taking the assignment that sets variables $x_{u,v}$ to true if $(u, v) \in M$ and to false otherwise. All pigeon axioms in $\mathcal{U}'$ correspond to matched pigeons and hence have a variable that is set to true. All hole axioms in E are satisfied because the only variables that were set to true follow the matching M and, hence, each hole has at most one of its neighbouring pigeons assigned to it. Thus $\mathcal{U}' \wedge E$ is satisfiable whenever $|\mathcal{U}'| \leq \epsilon n$.

As every $V_v \in \mathcal{V}$ has only variables that mention a single hole v, it holds that the overlap $ol(\mathcal{V}) = 1$. Applying Theorem 31 we get that any polynomial calculus resolution refutation of $PHP(G)$ requires degree at least $\epsilon\delta n - 1$. Applying Theorem 24 we get the exponential lower bound on the size of refuting $PHP(G)$, which proves our lemma. □

6 Empirical Results on SAT Solver Performance

For our experiments we used the SAT solvers Glucose 2.2 [Glu], March-rw [Mar], and Lingeling-ala [Lin]. The experiments were run under Linux on a computer with two quad-core AMD Opteron 2.2 GHz CPUs (2374 HE) and 16 GB of memory, where only one solver was running on the computer at any given time. We limited the solver running time to 1 hour per instance and the solvers access to memory was restricted to 8 GB.[6] For the experiments with fixed variable ordering we used a

[6]The reason is that each of two CPUs has its own primary memory bank, which is half of the memory available to the computer. Hence, relaxing the memory limit would allow access to the slower memory of the second CPU which would add more noise to the measurements.

version of MiniSat 2.2.0 [Min] modified so that the solver always branches on unset variable in fixed order.

The CNF formula instances were obtained as follows:

1. The formulas $SC^*(\sigma)$ from [Spe10] were generated by taking one fixed partition of $[4n+1]$ into $\{1,2,3,4\}$, $\{5,6,7,8\}$, ..., $\{4n-3,4n-2,4n-1,4n,4n+1\}$ and one random partition into 4-groups plus one 5-group, and then encoding positive and negative cardinality constraints, respectively, on these two partitions.

2. For the formulas $SC(G)$ from [VS10] we started with a random (non-bipartite) 4-regular graph, took the bipartite double cover (with two copies v_L, v_R of each vertex v and edges (u_L, v_R) for all edges (u,v) in the original graph), and finally added an additional edge.[7]

3. The fixed bandwidth formulas were constructed from an $n \times n$ matrix with ones in the first row on positions $1,2,4,8$ and zeroes everywhere else, and with every subsequent row being a cyclic shift one step to the right of the preceding row. Finally, an extra one was added to the top right cell of the matrix if this was a zero, and otherwise to the nearest cell containing a zero.[8]

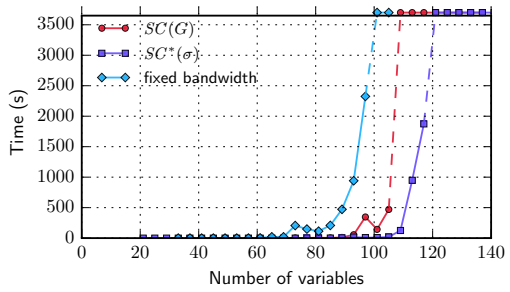
For each CNF formula we ran each SAT solver three times (with different random seeds), and for randomly generated formulas we ran on three different CNF formulas for each parameter value. Randomly generated formulas were generated with density 4.5. Out of the formulas that were solved by at least one of the solvers there were 389 unsatisfiable instances and 33 satisfiable. For formulas with at least 200 variables there were 280 unsatisfiable instances and 7 satisfiable, while the largest satisfiable instance had 357 variables. The values in the plots are the medians of these values. We also performed exactly the same set of experiments on randomly shuffled version of the formulas (with randomly permuted clauses, variables, and polarities), but this random shuffling did not affect the results in any significant way and so we do not display these plots.

We present the results of our experiments in Figure 3 with one subplot per solver.[9] As can be seen from these plots, all three versions of the formulas become infeasible for around 100–120 variables. Comparing to our experiments on random 3-CNF formulas and Tseitin formulas on random 3-regular graphs in Figure 4, it should be clear that all three flavours of the formulas from [Spe10, VS10] that we investigated were substantially harder than random formulas. Notice that for Tseitin formulas we do not present results for March and that Lingeling was run without

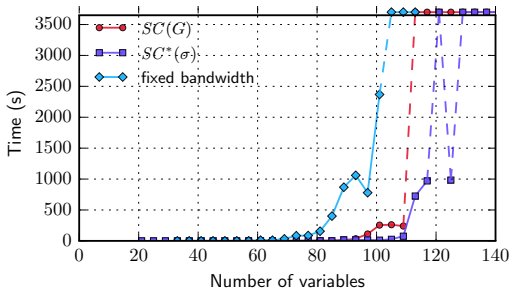
[7] We remark that, strictly speaking, this does not yield uniformly random instances but we just wanted to obtain some instances with "good enough" randomness (and hence expansion) on which we could run experiments.

[8] Note that this construction yields quadrangle-free instances for large enough n, except possibly for quadrangles involving the added extra top-right entry.

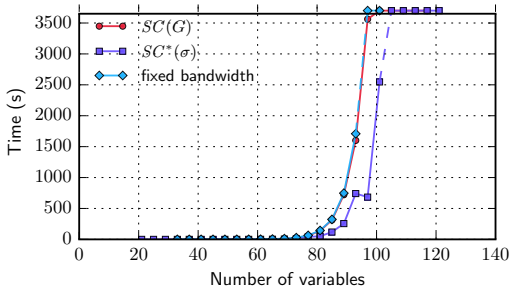
[9] The code for generating the CNF instances and complete data for the experiments can be found at `http://www.csc.kth.se/~jakobn/publications/sat14/`.



(a) Glucose

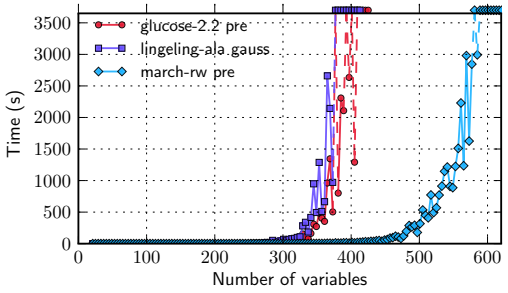


(b) Lingeling

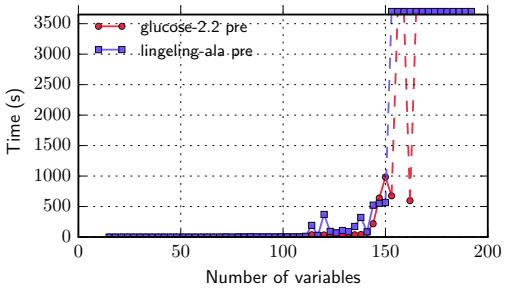


(c) March

Figure 3: SAT solver performance on variants of the formulas in [Spe10, VS10].



(a) Random 3-CNF formulas



(b) Tseitin formulas on random graphs

Figure 4: SAT solver performance on two well-known hard formula families.

Gaussian elimination. The reason is that March and Lingeling with Gaussian elimination solve Tseitin formulas in less than a second for even the largest instances we have tried.

Comparing random instances of formulas $SC^*(\sigma)$ from [Spe10] and $SC(G)$ from [VS10] with fixed bandwidth instances, we can see that the easiest ones are $SC^*(\sigma)$ while $SC(G)$ are somewhat harder. This is as expected—by construction, for formulas $SC(G)$ we are guaranteed that no pair of positive and negative constraints share more than one variable, whereas for formulas $SC^*(\sigma)$ it could happen in principle that a positive and a negative constraint act on two, three, or even four common variables. Somewhat counter-intuitively, however, the instances that are hardest in our practical experiments are the theoretically easy fixed bandwidth formulas.

In order to investigate whether the hardness of fixed bandwidth formulas could be attributed to hidden constants in the asymptotics—i.e., that the polynomial upper bounds on resolution length are so large in practice that the fixed bandwidth formulas are infeasible for all practical purposes—we ran a modified version of MiniSat on these formulas which always branched on variables row by row and in every row column by column. Intuitively, this seems to be the appropriate variable ordering if one is to recover the polynomial-length resolution refutation presented in [VS10]. And indeed, MiniSat run on fixed bandwidth formulas with fixed variable ordering performed much better than any of the other solvers on random instances of $SC^*(\sigma)$ and $SC(G)$ formulas. (We also verified that fixed variable ordering is not a good idea in general—as expected, MiniSat with fixed variable ordering performs poorly on random instances of $SC^*(\sigma)$ and $SC(G)$ formulas.) However, while our variable ordering gives faster running times for fixedbandwidth formulas, it can be significantly improved. Using a different ordering, as for instance one suggested by [Elf15], gives significantly better results than the ones produced by our ordering.

Given the latest advances in SAT solving technology, with solvers going beyond resolution by incorporating elements of algebraic reasoning (Gröbner bases) and geometric reasoning (pseudo-Boolean solvers), a natural question is whether the formulas in [Spe10, VS10] remain hard for such solvers.

Regarding algebraic solvers, we are not aware of any general-purpose solvers that can compete with CDCL solvers, but as mentioned the theoretical lower bounds that we prove for resolution hold also for polynomial calculus, which is a proof system for formalizing the reasoning in solvers based on Gröbner basis computations. Also, one can note that the algebraic reasoning in terms of Gaussian elimination in Lingeling does not seem to help.

For pseudo-Boolean solvers, which can be seen to search for proofs in (more or less restricted version of) the cutting planes proof system [CCT87], the story could potentially be very different. As noted multiple times already, the formulas $SC^*(\sigma)$ and $SC(G)$ are just encodings of a fairly simple counting principle, and in contrast to resolution and polynomial calculus the cutting planes proof system knows how to count. Thus, pseudo-Boolean solvers with enough well-developed methods of cardinality constraints reasoning should have the potential to solve these formulas quickly. This indeed appears to be the case as reported in [BLLM14], and our own (albeit limited) experiments also show this.

7 Concluding Remarks

In this work, we establish that the formulas constructed by Spence [Spe10] and Van Gelder and Spence [VS10] are exponentially hard for resolution and also for polynomial calculus resolution (PCR), which extends resolution with Gröbner basis computations. Formally, we prove that if the bipartite (multi-)graph describing the constraints encoded by the formula is expanding, then this implies exponential lower bounds on proof size in resolution and PCR. Furthermore, we show that

random instances of these formulas are almost surely expanding, meaning that the exponential lower bound applies with high probability.

We also investigate the performance of some current state-of-the-art SAT solvers on these formulas, and find that small instances are indeed much harder than, e.g., random 3-CNF formulas with the same number of variables. Somewhat surprisingly, however, the very hardest formulas in our experiments are versions of the formulas in [Spe10, VS10] generated from fixed bandwidth matrices. This is intriguing, since such formulas are easy for resolution, and since the current conventional wisdom (based on [AFT11, PD11]) seems to be that CDCL solvers can search efficiently for short resolution proofs. In view of this, an interesting (albeit very speculative) question is whether perhaps these fixed bandwidth matrix formulas could be used to show formally that CDCL with VSIDS, 1UIP, and phase saving, say, does *not* polynomially simulate resolution.

Since the formulas in [Spe10, VS10] encode what is in essence a fairly simple counting argument, SAT solvers that can reason efficiently with cardinality constraints could potentially solve these formulas fast. This indeed turns out to be the case for the latest version of Sat4j [BLLM14]. It would be interesting to investigate whether the formulas in [Spe10, VS10] could be slightly obfuscated to make them hard also for solvers with cardinality constraints. If so, this could yield small benchmark formulas that are hard not only for standard CDCL solvers but also for solvers extended with algebraic and/or geometric reasoning.

Another candidate construction of small but very hard CNF formulas is the one presented by Markström [Mar06]. It would be interesting to investigate what theoretical hardness results can be established for these formulas (for resolution and proof systems stronger than resolution) and how the practical hardness scales compared to the constructions by Spence and Van Gelder [Spe10, VS10]. In particular, an interesting question is whether these formulas, too, become easy for CDCL solvers enhanced with cardinality constraints reasoning.

Acknowledgements

The authors are very grateful to Massimo Lauria and Marc Vinyals for stimulating discussions and for invaluable practical help with setting up and evaluating the experiments. We wish to thank Niklas Sörensson for explaining how to fix the variable decision order in MiniSat, and Daniel Le Berre for sharing data about the performance of the latest version of Sat4j on our benchmark formulas. We are also grateful to Allen Van Gelder for comments on a preliminary write-up of some of the results in this paper, as well as for introducing us to this problem in the first place. Finally, we thank several participants of the workshop *Theoretical Foundations of Applied SAT Solving (14w5101)* at the Banff International Research Station in January 2014 for interesting conversations on themes related to this work.

The authors were funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement

no. 279611. The second author was also supported by the Swedish Research Council grants 621-2010-4797 and 621-2012-5645.

Bibliography

[ABRW02] 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*.

[AFT11] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *Journal of Artificial Intelligence Research*, 40:353–373, January 2011. Preliminary version in *SAT '09*.

[AR03] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. *Proceedings of the Steklov Institute of Mathematics*, 242:18–35, 2003. Available at `http://people.cs.uchicago.edu/~razborov/files/misha.pdf`. Preliminary version in *FOCS '01*.

[BI13] Chris Beck and Russell Impagliazzo. Strong ETH holds for regular resolution. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC '13)*, pages 487–494, May 2013.

[Bla37] Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.

[BLLM14] Armin Biere, Daniel Le Berre, Emmanuel Lonca, and Norbert Manthey. Detecting cardinality constraints in CNF. In *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *Lecture Notes in Computer Science*, pages 285–301. Springer, July 2014.

[BM08] John Adrian Bondy and Uppaluri Siva Ramachandra Murty. *Graph Theory*. Springer, 2008.

[BS97] Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pages 203–208, July 1997.

[BS13] Jose Blanchet and Alexandre Stauffer. Characterizing optimal sampling of binary contingency tables via the configuration model. *Random Structures & Algorithms*, 42(2):159–184, 2013.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

[CCT87] William Cook, Collette Rene Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, November 1987.

[CEI96] 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)*, pages 174–183, May 1996.

[CS88] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[Elf15] Jan Elffers. Personal communication, 2015.

[Glu] The Glucose SAT solver. `http://www.labri.fr/perso/lsimon/glucose/`.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, October 2006.

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[Jan13] Svante Janson. The probability that a random multigraph is simple, ii. Technical Report 1307.6344, arXiv.org, 2013. Available at `http://arxiv.org/abs/1307.6344`.

[Lin] Lingeling and Plingeling. `http://fmv.jku.at/lingeling/`.

[LP10] Daniel Le Berre and Anne Parrain. The Sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation*, 7:59–64, 2010.

[Mar] March. `http://www.st.ewi.tudelft.nl/~marijn/sat/march\_dl.php`.

[Mar06] Klas Markström. Locality and hard SAT-instances. *Journal on Satisfiability, Boolean Modeling and Computation*, 2(1-4):221–227, 2006.

[Min] The MiniSat page. `http://minisat.se/`.

[MMZ+01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC '01)*, pages 530–535, June 2001.

[MN14] Mladen Mikša and Jakob Nordström. Long proofs of (seemingly) simple formulas. In *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *Lecture Notes in Computer Science*, pages 121–137. Springer, July 2014.

[MN15] Mladen Mikša and Jakob Nordström. A generalized method for proving polynomial calculus degree lower bounds. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 467–487, June 2015.

[MS99] João P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, May 1999. Preliminary version in *ICCAD '96*.

[Nor13] Jakob Nordström. Pebble games, proof complexity and time-space tradeoffs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

[PD11] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artificial Intelligence*, 175:512–525, February 2011. Preliminary version in *CP '09*.

[Spe10] Ivor Spence. sgen1: A generator of small but difficult satisfiability benchmarks. *Journal of Experimental Algorithmics*, 15:1.2:1.1–1.2:1.15, March 2010.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

[VS10] Allen Van Gelder and Ivor Spence. Zero-one designs produce small hard SAT instances. In *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT '10)*, volume 6175 of *Lecture Notes in Computer Science*, pages 388–397. Springer, July 2010.

D

A Generalized Method for Proving Polynomial Calculus Degree Lower Bounds*

Mladen Mikša[1] and Jakob Nordström[1]

[1]KTH Royal Institute of Technology

Abstract

We study the problem of obtaining lower bounds for polynomial calculus (PC) and polynomial calculus resolution (PCR) on proof degree, and hence by [Impagliazzo et al. '99] also on proof size. [Alekhnovich and Razborov '03] established that if the clause-variable incidence graph of a CNF formula F is a good enough expander, then proving that F is unsatisfiable requires high PC/PCR degree. We further develop the techniques in [AR03] to show that if one can "cluster" clauses and variables in a way that "respects the structure" of the formula in a certain sense, then it is sufficient that the incidence graph of this clustered version is an expander. As a corollary of this, we prove that the functional pigeonhole principle (FPHP) formulas require high PC/PCR degree when restricted to constant-degree expander graphs. This answers an open question in [Razborov '02], and also implies that the standard CNF encoding of the FPHP formulas require exponential proof size in polynomial calculus resolution. Thus, while Onto-FPHP formulas are easy for polynomial calculus, as shown in [Riis '93], both FPHP and Onto-PHP formulas are hard even when restricted to bounded-degree expanders.

1 Introduction

In one sentence, proof complexity studies how hard it is to certify the unsatifiability of formulas in conjunctive normal form (CNF). In its most general form, this is the question of whether coNP can be separated from NP or not, and as such it still appears almost completely out of reach. However, if one instead focuses on concrete proof systems, which can be thought of as restricted models of (nondeterministic) computation, then fruitful study is possible.

*This is an updated and strengthened full-length version of the paper [MN15], which appeared in the *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*.

Resolution and Polynomial Calculus

Perhaps the most well-studied proof system is *resolution* [Bla37], in which one derives new disjunctive clauses from a CNF formula until an explicit contradiction is reached, and for which numerous exponential lower bounds on proof size have been shown (starting with [Hak85, Urq87, CS88]). Many of these lower bounds can be established by instead studying the *width* of proofs, i.e., the size of a largest clause appearing in the proofs, and arguing that any resolution proof for a certain formula must contain a large clause. It then follows from a result by Ben-Sasson and Wigderson [BW01] that any resolution proof must also consist of very many clauses. Research since [BW01] has led to a well-developed machinery for showing width lower bounds, and hence also size lower bounds.

The focus of the current paper is the slightly more general proof system *polynomial calculus resolution (PCR)*. This proof system was introduced by Clegg et al. [CEI96] in a slightly weaker form that is usually referred to as *polynomial calculus (PC)* and was later extended by Alekhnovich et al. [ABRW02]. In PC and PCR clauses are translated to multilinear polynomials over some (fixed) field $\mathbb{F}$, and a CNF formula F is shown to be unsatisfiable by proving that the constant 1 lies in the ideal generated by the polynomials corresponding to the clauses of F. Here the size of a proof is measured as the number of monomials in a proof when all polynomials are expanded out as linear combinations of monomials, and the width of a clause corresponds to the (total) *degree* of the polynomial representing the clause. Briefly, the difference between PC and PCR is that the latter proof system has separate formal variables for positive and negative literals over the same variable. Thanks to this, one can encode wide clauses into polynomials compactly regardless of the sign of the literals in the clauses, which allows PCR to simulate resolution efficiently. With respect to the degree measure polynomial calculus and polynomial calculus resolution are exactly the same, and furthermore the degree needed to prove in polynomial calculus that a formula is unsatisfiable is at most the width required in resolution.

In a work that served, interestingly enough, as a precursor to [BW01], Impagliazzo et al. [IPS99] showed that strong lower bounds on the degree of PC proofs are sufficient to establish strong size lower bounds. The same proof goes through for PCR, and hence any lower bound on proof size obtained via a degree lower bound applies to both PC and PCR. In this paper, we will therefore be somewhat sloppy in distinguishing the two proof systems, sometimes writing "polynomial calculus" to refer to both systems when the results apply to both PC and PCR.

In contrast to the situation for resolution after [BW01], the paper [IPS99] has not been followed by a corresponding development of a generally applicable machinery for proving degree lower bounds. For fields of characteristic distinct from 2 it is sometimes possible to obtain lower bounds by doing an affine transformation from $\{0, 1\}$ to the "Fourier basis" $\{-1, +1\}$, an idea that seems to have appeared first in [Gri98, BGIP01]. For fields of arbitrary characteristic Alekhnovich and Razborov [AR03] developed a powerful technique for general systems of polynomial

equations, which when restricted to the standard encoding of CNF formulas F yields that polynomial calculus proofs require high degree if the corresponding bipartite clause-variable incidence graphs $G(F)$ are good enough expanders. There are many formula families for which this is not true, however. One can have a family of constraint satisfaction problems where the constraint-variable incidence graph is an expander—say, for instance, for an unsatisfiable set of linear equations mod 2—but where each constraint is then translated into several clauses when encoded into CNF, meaning that the clause-variable incidence graph $G(F)$ will no longer be expanding. For some formulas this limitation is inherent—it is not hard to see that an inconsistent system of linear equations mod 2 is easy to refute in polynomial calculus over $\mathbb{F}_2$, and so good expansion for the constraint-variable incidence graph should *not* in itself be sufficient to imply hardness in general—but in other cases it would seem that some kind of expansion of this sort should still be enough, "morally speaking," to guarantee that the corresponding CNF formulas are hard.[1]

Pigeonhole Principle Formulas

One important direction in proof complexity, which is the reason research in this area was initiated by Cook and Reckhow [CR79], has been to prove superpolynomial lower bounds on proof size for increasingly stronger proof systems. For proof systems where such lower bounds have already been obtained, however, such as resolution and polynomial calculus, a somewhat orthogonal research direction has been to try to gain a better understanding of the strengths and weaknesses of a given proof system by studying different combinatorial principles (encoded in CNF) and determining how hard they are to prove for this proof system.

It seems fair to say that by far the most extensively studied such combinatorial principle is the *pigeonhole principle*. This principle is encoded into CNF as unsatisfiable formulas claiming that m pigeons can be mapped in a one-to-one fashion into n holes for $m > n$, but there are several choices exactly how to do this encoding. The most basic *pigeonhole principle (PHP) formulas* have clauses saying that every pigeon gets at least one pigeonhole and that no hole contains two pigeons. While these formulas are already unsatisfiable for $m \geq n + 1$, they do not a priori rule out that there might be "fat" pigeons residing in several holes. The *functional pigeonhole principle (FPHP) formulas* perhaps correspond more closely to our intuitive understanding of the pigeonhole principle in that they also contain

[1] In a bit more detail, what is shown in [AR03] is that if the constraint-variable incidence graph for a set of polynomial equations is a good expander, and if these polynomials have high immunity—i.e., do not imply other polynomials of significantly lower degree—then proving that this set of polynomial equations is inconsistent in polynomial calculus requires high degree. CNF formulas automatically have maximal immunity since a clause translated into a polynomial does not have any consequences of degree lower than the width of the clause in question, and hence expansion of the clause-variable incidence graph is sufficient to imply hardness for polynomial calculus. Any polynomial encoding of a linear equation mod 2 has a low-degree consequence over $\mathbb{F}_2$, however—namely, the linear equation itself—and this is why [AR03] (correctly) fails to prove lower bounds in this case.

functionality clauses specifying that every pigeon gets exactly one pigeonhole and not more. Another way of making the basic PHP formulas more constrained is to add *onto* clauses requiring that every pigeonhole should get a pigeon, yielding so-called *onto-PHP formulas*. Finally, the most restrictive encoding, and hence the hardest one when it comes to proving lower bounds, are the *onto-FPHP formulas* containing both functionality and onto clauses, i.e., saying that the mapping from pigeons to pigeonholes is a perfect matching. Razborov's survey [Raz02] gives a detailed account of these different flavours of the pigeonhole principle formulas and results for them with respect to various proof systems—we just quickly highlight some facts relevant to this paper below.

For the resolution proof system there is not much need to distinguish between the different PHP versions discussed above. The lower bound by Haken [Hak85] for formulas with $m = n + 1$ pigeons can be made to work also for onto-FPHP formulas, and more recent works by Raz [Raz04a] and Razborov [Raz03, Raz04b] show that the formulas remain exponentially hard (measured in the number of pigeonholes n) even for arbitrarily many pigeons m.

Interestingly enough, for polynomial calculus the story is very different. The first degree lower bounds were proven by Razborov [Raz98], but for a different encoding than the standard translation from CNF, since translating wide clauses yields initial polynomials of high degree. Alekhnovich and Razborov [AR03] proved lower bounds for a 3-CNF version of the pigeonhole principle, from which it follows that the standard CNF encoding requires proofs of exponential size. However, as shown by Riis [Rii93] the onto-FPHP formulas with $m = n + 1$ pigeons are easy for polynomial calculus. And while the encoding in [Raz98] also captures the functionality restriction in some sense, it has remained open whether the standard CNF encoding of functional pigeonhole principle formulas translated to polynomials is hard (this question has been highlighted, for instance, in Razborov's open problems list [Raz15]).

Another way of modifying the pigeonhole principle is to restrict the choices of pigeonholes for each pigeon by defining the formulas over a bipartite graph $H = (U \dot\cup V, E)$ with $|U| = m$ and $|V| = n$ and requiring that each pigeon $u \in U$ goes to one of its neighbouring holes in $N(u) \subseteq V$. If the graph H has constant left degree, the corresponding *graph pigeonhole principle formula* has constant width and a linear number of variables, which makes it possible to apply [BW01, IPS99] to obtain exponential proof size lower bounds from linear width/degree lower bounds. A careful reading of the proofs in [AR03] reveals that this paper establishes linear polynomial calculus degree lower bounds (and hence exponential size lower bounds) for graph PHP formulas, and in fact also graph Onto-PHP formulas, over constant-degree expanders H. Razborov lists as one of the open problems in [Raz02] whether this holds also for graph FPHP formulas, i.e., with functionality clauses added, from which exponential lower bounds on polynomial calculus proof size for the general FPHP formulas would immediately follow.

Our Results

We revisit the technique developed in [AR03] for proving polynomial calculus degree lower bounds, restricting our attention to the special case when the polynomials are obtained by the canonical translation of CNF formulas.

Instead of considering the standard clause-variable incidence graph $G(F)$ of a CNF formula F (with clauses on the left, variables on the right, and edges encoding that a variable occurs in a clause) we construct a new graph G' by clustering several clauses and/or variables into single vertices, reflecting the structure of the combinatorial principle the CNF formula F is encoding. The edges in this new graph G' are the ones induced by the original graph $G(F)$ in the natural way, i.e., there is an edge from a left cluster to a right cluster in G' if any clause in the left cluster has an edge to any variable in the right cluster in $G(F)$. We remark that such a clustering is already implicit in, for instance, the resolution lower bounds in [BW01] for Tseitin formulas (which is essentially just a special form of unsatisfiable linear equations) and graph PHP formulas, as well as in the graph PHP lower bound for polynomial calculus in [AR03].

We then show that if this clustering is done in the right way, the proofs in [AR03] still go through and yield strong polynomial calculus degree lower bounds when G' is a good enough expander.[2] It is clear that this cannot work in general—as already discussed above, any inconsistent system of linear equations $\bmod 2$ is easy to refute in polynomial calculus over $\mathbb{F}_2$, even though for a random instance of this problem the clauses encoding each linear equation can be clustered to yield an excellent expander G'. Very informally (and somewhat incorrectly) speaking, the clustering should be such that if a cluster of clauses F' on the left is a neighbour of a variable cluster V on the right, then there should exist an assignment ρ to V such that ρ satisfies all of F' and such that for the clauses outside of F' they are either satisfied by ρ or left completely untouched by ρ. Also, it turns out to be helpful not to insist that the clustering of variables on the right should be a partition, but that we should allow the same variable to appear in several clusters if needed (as long as the number of clusters for each variable is bounded).

This extension of the lower bound method in [AR03] makes it possible to present previously obtained polynomial calculus degree lower bounds in [AR03, GL10, MN14] in a unified framework. Moreover, it allows us to prove the following new results:

1. If a bipartite graph $H = (U \dot\cup V, E)$ with $|U| = m$ and $|V| = n$ is a boundary expander (a.k.a. unique-neighbour expander), then the graph FPHP formula over H requires proofs of linear polynomial calculus degree, and hence exponential polynomial calculus size.

2. Since FPHP formulas can be turned into graph FPHP formulas by hitting them with a restriction, and since restrictions can only decrease proof size, it

[2] For a certain twist of the definition of expander that we do not describe in full detail here in order to keep the discussion at an informal, intuitive level. The formal description is given in Section 3.

follows that FPHP formulas require proofs of exponential size in polynomial calculus.

This fills in the last missing pieces in our understanding of the different flavours of pigeonhole principle formulas with $n+1$ pigeons and n holes for polynomial calculus. Namely, while Onto-FPHP formulas are easy for polynomial calculus, both FPHP formulas and Onto-PHP formulas are hard even when restricted to expander graphs.

Organization of This Paper

After reviewing the necessary preliminaries in Section 2, we present our extension of the Alekhnovich–Razborov method in Section 3. In Section 4, we show how this method can be used to rederive some previous polynomial calculus degree lower bounds as well as to obtain new degree and size lower bounds for functional (graph) PHP formulas. We conclude in Section 5 by discussing some possible directions for future research.

2 Preliminaries

Let us start by giving an overview of the relevant proof complexity background. This material is standard and we refer to, for instance, the survey [Nor13] for more details.

A *literal* over a Boolean variable x is either the variable x itself (a *positive literal*) or its negation $\neg x$ or $\overline{x}$ (a *negative literal*). We define $\overline{\overline{x}} = x$. We identify 0 with true and 1 with false. We remark that this is the opposite of the standard convention in proof complexity, but it is a more natural choice in the context of polynomial calculus, where "evaluating to true" means "vanishing." A *clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of literals. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses. The *width* $W(C)$ of a clause C is the number of literals $|C|$ in it, and the width $W(F)$ of the formula F is the maximum width of any clause in the formula. We think of clauses and CNF formulas as sets, so that order is irrelevant and there are no repetitions. A k-CNF formula has all clauses of size at most k, where k is assumed to be some fixed constant.

In polynomial calculus resolution the goal is to prove the unsatisfiability of a CNF formula by reasoning with polynomials from a polynomial ring $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$ (where x and $\overline{x}$ are viewed as distinct formal variables) over some fixed field $\mathbb{F}$. The results in this paper hold for all fields $\mathbb{F}$ regardless of characteristic. In what follows, a *monomial* m is a product of variables and a *term* t is a monomial multiplied by an arbitrary non-zero field element.

Definition 1 (Polynomial calculus resolution (PCR) [CEI96, ABRW02]). A *polynomial calculus resolution (PCR) refutation* $\pi : F \vdash \bot$ of a CNF formula F (also referred to as a *PCR proof* for F) over a field $\mathbb{F}$ is an ordered sequence of

polynomials $\pi = (P_1, \ldots, P_\tau)$, expanded out as linear combinations of monomials, such that $P_\tau = 1$ and each line P_i, $1 \leq i \leq \tau$, is either

- a monomial $\prod_{x \in L^+} x \cdot \prod_{y \in L^-} \overline{y}$ encoding a clause $\bigvee_{x \in L^+} x \vee \bigvee_{y \in L^-} \overline{y}$ in F (a *clause axiom*);
- a *Boolean axiom* $x^2 - x$ or *complementarity axiom* $x + \overline{x} - 1$ for any variable x;
- a polynomial obtained from one or two previous polynomials in the sequence by *linear combination* $\frac{Q \quad R}{\alpha Q + \beta R}$ or *multiplication* $\frac{Q}{xQ}$ for any $\alpha, \beta \in \mathbb{F}$ and any variable x.

If we drop complementarity axioms and encode each negative literal $\overline{x}$ as $(1 - x)$, the proof system is called *polynomial calculus (PC)*.

The *size* $S(\pi)$ of a PC/PCR refutation $\pi = (P_1, \ldots, P_\tau)$ is the number of monomials in π (counted with repetitions),[3] the *degree* $Deg(\pi)$ is the maximal degree of any monomial appearing in π, and the *length* $L(\pi)$ is the number τ of polynomials in π. Taking the minimum over all PCR refutations of a formula F, we define the size $S_{\mathcal{PCR}}(F \vdash \bot)$, degree $Deg_{\mathcal{PCR}}(F \vdash \bot)$, and length $L_{\mathcal{PCR}}(F \vdash \bot)$ of refuting F in PCR (and analogously for PC).

We write $Vars(C)$ and $Vars(m)$ to denote the set of all variables appearing in a clause C or monomial (or term) m, respectively and extend this notation to CNF formulas and polynomials by taking unions. We use the notation $\langle P_1, \ldots, P_m \rangle$ for the ideal generated by the polynomials P_i, $i \in [m]$. That is, $\langle P_1, \ldots, P_m \rangle$ is the minimal subset of polynomials containing all P_i that is closed under addition and multiplication by any polynomial. One way of viewing a polynomial calculus (PC or PCR) refutation is as a calculation in the ideal generated by the encodings of clauses in F and the Boolean and complementarity axioms. It can be shown that such an ideal contains 1 if and only if F is unsatisfiable.

As mentioned above, we have $Deg_{\mathcal{PCR}}(F \vdash \bot) = Deg_{\mathcal{PC}}(F \vdash \bot)$ for any CNF formula F. This claim can essentially be verified by taking any PCR refutation of F and replacing all occurrences of $\overline{y}$ by $(1 - y)$ to obtain a valid PC refutation in the same degree. Hence, we can drop the subscript from the notation for the degree measure. We have the following relation between refutation size and refutation degree (which was originally proven for PC but the proof of which also works for PCR).

Theorem 2 ([IPS99]). *Let F be an unsatisfiable CNF formula of width $W(F)$ over n variables. Then*

$$S_{\mathcal{PCR}}(F \vdash \bot) = \exp\left(\Omega\left(\frac{(Deg(F \vdash \bot) - W(F))^2}{n}\right)\right) \quad .$$

[3] We remark that the natural definition of size is to count monomials with repetition, but all lower bound techniques known actually establish slightly stronger lower bounds on the number of *distinct* monomials.

Thus, for k-CNF formulas it is sufficient to prove strong enough lower bounds on the PC degree of refutations to establish strong lower bounds on PCR proof size.

Furthermore, it will be convenient for us to simplify the definition of PC so that axioms $x^2 - x$ are always applied implicitly whenever possible. We do this by defining the result of the multiplication operation to be the multilinearized version of the product. This can only decrease the degree (and size) of the refutation, and is in fact how polynomial calculus is defined in [AR03]. Hence, from now on whenever we refer to polynomials and monomials we mean multilinear polynomials and multilinear monomials, respectively, and polynomial calculus is defined over the (multilinear) polynomial ring $\mathbb{F}[x, y, z, \ldots]/\langle x^2 - x, y^2 - y, z^2 - z, \ldots\rangle$.

It might be worth noticing that for this modified definition of polynomial calculus it holds that any (unsatisfiable) k-CNF formula can be refuted in linear length (and hence, in constrast to resolution, the size of refutations, rather than the length, is the right measure to focus on). This is not hard to show, and in some sense is probably folklore, but since it does not seem to be too widely known we state it for the record and provide a proof.

Proposition 3. *Any unsatisfiable k-CNF formula F has a (multilinear) polynomial calculus refutation of length linear in the size of the formula F.*

Proof. We show by induction how to derive polynomials $P_j = 1 - \prod_{i=1}^{j}(1 - C_i)$ in length linear in j, where we identify the clause C_i in $F = \bigwedge_{i=1}^{m} C_i$ with the polynomial encoding of this clause. The end result is the polynomial $P_m = 1 - \prod_{i=1}^{m}(1 - C_i)$. As F is unsatisfiable, for every 0-1 assignment there is at least one C_i that evaluates to 1 and hence P_m evaluates to 1. Thus, P_m is equal to 1 on all 0-1 assignments. However, it is a basic fact that every function $f : \{0, 1\}^n \to \mathbb{F}$ is uniquely representable as a multilinear polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ (since the multilinear monomials span this vector space and are linearly independent, they form a basis). Therefore, it follows that P_m is syntactically equal to the polynomial 1.

The base case of the induction is the polynomial P_1 that is equal to C_1. To prove the induction step, we need to show how to derive

$$P_{j+1} = 1 - \prod_{i=1}^{j+1}(1 - C_i) = 1 - (1 - C_{j+1})(1 - P_j) = P_j + C_{j+1} - C_{j+1}P_j \quad (2.1)$$

from P_j and C_{j+1} in a constant number of steps. To start, we derive $C_{j+1}P_j$ from P_j, which can be done with a constant number of multiplications and additions since the width/degree of C_{j+1} is upper-bounded by the constant k. We derive P_{j+1} in two more steps by first taking a linear combination of P_j and $C_{j+1}P_j$ to get $P_j - C_{j+1}P_j$ and then adding C_{j+1} to this to obtain $P_j - C_{j+1}P_j + C_{j+1} = P_{j+1}$. The proposition follows. □

We will also need to use restrictions. A *restriction* ρ on F is a partial assignment to the variables of F. We use $\mathrm{Dom}(\rho)$ to denote the set of variables assigned by ρ. In a restricted formula $F\restriction_\rho$ all clauses satisfied by ρ are removed and all other

clauses have falsified literals removed. For a PC refutation π restricted by ρ we have that if ρ satisfies a literal in a monomial, then that monomial is set to 0 and vanishes, and all falsified literals in a monomial get replaced by 1 and disappear. It is not hard to see that if π is a PC (or PCR) refutation of F, then $\pi{\restriction_\rho}$ is a PC (or PCR) refutation of $F{\restriction_\rho}$, and this restricted refutation has at most the same size, degree, and length as the original refutation.

3 A Generalization of the Alekhnovich–Razborov Method for CNFs

Many lower bounds in proof complexity are proved by arguing in terms of expansion. One common approach is to associate a bipartite graph $G(F)$ with the CNF formula F with clauses on one side and variables on the other and with edges encoding that a variable occurs in a clause (the so-called *clause-variable incidence graph* mentioned in the introduction). The method we present below, which is an extension of the techniques developed by Alekhnovich and Razborov [AR03] (but restricted to the special case of CNF formulas), is a variation on this theme. As already discussed, however, we will need a slightly more general graph construction where clauses and variables can be grouped into clusters. We begin by describing this construction.

A Generalized Clause-Variable Incidence Graph

The key to our construction of generalized clause-variable incidence graphs is to keep track of how clauses in a CNF formula are affected by partial assignments.

Definition 4 (Respectful assignments and variable sets). We say that a partial assignment ρ *respects* a CNF formula E, or that ρ is *E-respectful*, if for every clause C in E either $\mathit{Vars}(C) \cap \mathrm{Dom}(\rho) = \emptyset$ or ρ satisfies C. A set of variables V respects a CNF formula E if there exists an assignment ρ with $\mathrm{Dom}(\rho) = V$ that respects E.

Example 5. Consider the CNF formula $E = (x_1 \wedge x_2) \wedge (\overline{x}_1 \wedge x_3) \wedge (x_1 \wedge x_4) \wedge (\overline{x}_1 \wedge x_5)$ and the subsets of variables $V_1 = \{x_1, x_2, x_3\}$ and $V_2 = \{x_4, x_5\}$. The assignment ρ_2 to V_2 setting x_4 and x_5 to true respects E since it satisfies the clauses containing these variables, and hence V_2 is E-respectful. However, V_1 is not E-respectful since setting x_1 will affect all clauses in E but cannot satisfy both $x_1 \wedge x_4$ and $\overline{x}_1 \wedge x_5$.

Definition 6 (Respectful satisfaction). Let F and E be CNF formulas and let V be a set of variables. We say that F is *E-respectfully satisfiable by V* if there exists a partial assignment ρ with $\mathrm{Dom}(\rho) = V$ that satisfies F and respects E. Such an assignment ρ is said to *E-respectfully satisfy F*.

Using a different terminology, Definition 4 says that ρ is an *autarky* for E, meaning that ρ satisfies all clauses in E which it touches, i.e., that $E{\restriction_\rho} \subseteq E$ after we

remove all satisfied clauses in $E|_\rho$. Definition 6 ensures that the autarky ρ satisfies the formula F.

Recall that we identify a CNF formula $\bigwedge_{i=1}^{m} C_i$ with the set of clauses $\{C_i \mid i \in [m]\}$. In the rest of this section we will switch freely between these two perspectives. We also change to the notation $\mathcal{F}$ for the input CNF formula, to free up other letters that will be needed in notation introduced below.

To build a bipartite graph representing the CNF formula $\mathcal{F}$, we will group the formula into subformulas (i.e., subsets of clauses). In what follows, we write $\mathcal{U}$ to denote the part of $\mathcal{F}$ that will form the left vertices of the constructed bipartite graph, while E denotes the part of $\mathcal{F}$ which will not be represented in the graph but will be used to enforce respectful satisfaction. In more detail, $\mathcal{U}$ is a family of subformulas F of $\mathcal{F}$ where each subformula is one vertex on the left-hand side of the graph. We also consider the variables of $\mathcal{F}$ to be divided into a family $\mathcal{V}$ of subsets of variables V. In our definition, $\mathcal{U}$ and $\mathcal{V}$ do not need to be partitions of clauses and variables in $\mathcal{F}$, respectively. This is not too relevant for $\mathcal{U}$ because we will always define it as a partition, but it turns out to be useful in our applications to have sets in $\mathcal{V}$ share variables. The next definition describes the bipartite graph that we build and distinguishes between two types of neighbour relations in this graph.

Definition 7 (Bipartite $(\mathcal{U},\mathcal{V})_E$-graph). Let E be a CNF formula, $\mathcal{U}$ be a set of CNF formulas, and $\mathcal{V}$ be a family of sets of variables V that respect E. Then the *(bipartite) $(\mathcal{U},\mathcal{V})_E$-graph* is a bipartite graph with left vertices $F \in \mathcal{U}$, right vertices $V \in \mathcal{V}$, and edges between F and V if $\mathit{Vars}(F) \cap V \neq \emptyset$. For every edge (F,V) in the graph we say that F and V are *E-respectful neighbours* if F is E-respectfully satisfiable by V. Otherwise, they are *E-disrespectful neighbours*.

We will often write $(\mathcal{U},\mathcal{V})_E$ as a shorthand for the graph defined by $\mathcal{U}$, $\mathcal{V}$, and E as above. We will also use standard graph notation and write $N(F)$ to denote the set of all neighbours $V \in \mathcal{V}$ of a vertex/CNF formula $F \in \mathcal{U}$. It is important to note that the fact that F and V are E-respectful neighbours can be witnessed by an assignment that falsfies other subformulas $F' \in \mathcal{U} \setminus \{F\}$.

We can view the formation of the $(\mathcal{U},\mathcal{V})_E$-graph as taking the clause-variable incidence graph $G(\mathcal{F})$ of the CNF formula $\mathcal{F}$, throwing out a part of $\mathcal{F}$, which we denote E, and clustering the remaining clauses and variables into $\mathcal{U}$ and $\mathcal{V}$. The edge relation in the $(\mathcal{U},\mathcal{V})_E$-graph follows naturally from this view, as we put an edge between two clusters if there is an edge between any two elements of these clusters. The only additional information we need to keep track of is which clause and variable clusters are E-respectful neighbours or not.

Definition 8 (Respectful boundary). For a $(\mathcal{U},\mathcal{V})_E$-graph and a subset $\mathcal{U}' \subseteq \mathcal{U}$, the *$E$-respectful boundary* $\partial_E(\mathcal{U}')$ of $\mathcal{U}'$ is the family of variable sets $V \in \mathcal{V}$ such that each $V \in \partial_E(\mathcal{U}')$ is an E-respectful neighbour of some clause set $F \in \mathcal{U}'$ but is not a neighbour (respectful or disrespectful) of any other clause set $F' \in \mathcal{U}' \setminus \{F\}$.

It will sometimes be convenient to interpret subsets $\mathcal{U}' \subseteq \mathcal{U}$ as CNF formulas $\bigwedge_{F \in \mathcal{U}'} \bigwedge_{C \in F} C$, and we will switch back and forth between these two interpretations as seems most suitable. We will show that a formula $\mathcal{F} = \bigwedge_{F \in \mathcal{U}} \bigwedge_{C \in F} C \wedge E = \mathcal{U} \wedge E$ is hard for polynomial calculus with respect to degree if the $(\mathcal{U}, \mathcal{V})_E$-graph has a certain expansion property as defined next.

Definition 9 (Respectful boundary expander). A $(\mathcal{U}, \mathcal{V})_E$-graph is said to be an (s, δ, ξ, E)-*respectful boundary expander*, or just an (s, δ, ξ, E)-*expander* for brevity, if for every set $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$, it holds that $|\partial_E(\mathcal{U}')| \geq \delta|\mathcal{U}'| - \xi$.

Note that an (s, δ, ξ, E)-respectful boundary expander is a standard bipartite boundary expander except for two modifications:

- We measure expansion not in terms of the whole boundary but only in terms of the *respectful boundary*[4] as described in Definition 8.
- Also, the size of the boundary $|\partial_E(\mathcal{U}')|$ on the right does not quite have to scale linearly with the size of the vertex set $|\mathcal{U}'|$ on the left. Instead, we allow an *additive loss* ξ in the expansion. In our applications, we can usually construct graphs with good enough expansion so that we can choose $\xi = 0$, but for one of the results we present it will be helpful to allow a small slack here.

Before we state our main theorem we need one more technical definition, which is used to ensure that there do not exist variables that appear in too many variable sets in $\mathcal{V}$. We remark that the concept below is also referred to as the "maximum degree" in the literature, but since we already have degrees of polynomials and vertices in this paper we prefer a new term instead of overloading "degree" with a third meaning.

Definition 10. The *overlap* of a variable x with respect to a family of variable sets $\mathcal{V}$ is $ol(x, \mathcal{V}) = |\{V \in \mathcal{V} : x \in V\}|$ and the overlap of $\mathcal{V}$ is $ol(\mathcal{V}) = \max_x\{ol(x, \mathcal{V})\}$, i.e., the maximum number of sets $V \in \mathcal{V}$ containing any particular variable x.

Given the above definitions, we can state the main technical result in this paper as follows.

Theorem 11. *Let $\mathcal{F} = \bigwedge_{F \in \mathcal{U}} \bigwedge_{C \in F} C \wedge E = \mathcal{U} \wedge E$ be a CNF formula for which $(\mathcal{U}, \mathcal{V})_E$ is an (s, δ, ξ, E)-expander with overlap $ol(\mathcal{V}) = d$, and suppose furthermore*

[4]Somewhat intriguingly, we will not see any disrespectful neighbours in our applications in Section 4, but the concept of respectfulness is of crucial importance for the main technical result in Theorem 11 to go through. One way of seeing this is to construct a $(\mathcal{U}, \mathcal{V})_E$-graph for an expanding set of linear equations mod 2, where $\mathcal{U}$ consists of the (CNF encodings of) the equations, $\mathcal{V}$ consists of one variable set for each equation containing exactly the variables in this equation, and E is empty. Then this $(\mathcal{U}, \mathcal{V})_E$-graph has the same boundary expansion as the constraint-variable incidence graph, but Theorem 11 does not apply (which it should not do) since this expansion is not respectful.

that for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$, it holds that $\mathcal{U}' \wedge E$ is satisfiable. Then any polynomial calculus refutation of $\mathcal{F}$ requires degree strictly greater than $(\delta s - 2\xi)/(2d)$.

In order to prove this theorem, it will be convenient to review some algebra. We do so next.

Some Algebra Basics

We will need to compute with polynomials modulo ideals, and in order to do so we need to have an ordering of monomials (which, as we recall, will always be multilinear).

Definition 12 (Admissible ordering). We say that a total ordering $\prec$ on the set of all monomials over some fixed set of variables is *admissible* if the following conditions hold:

- If $Deg(m_1) < Deg(m_2)$, then $m_1 \prec m_2$.
- For any m_1, m_2, and m such that m does not share any variables with m_1 or m_2 and $m_1 \prec m_2$, it holds that $mm_1 \prec mm_2$.

Two terms $t_1 = \alpha_1 m_1$ and $t_2 = \alpha_2 m_2$ are ordered in the same way as their underlying monomials m_1 and m_2.

One example of an admissible ordering is to first order monomials with respect to their degree and then lexicographically. For the rest of this section we only need that $\prec$ is some fixed but arbitrary admissible ordering, but the reader can think of the degree-lexicographical ordering without any particular loss of generality. We write $m_1 \preccurlyeq m_2$ to denote that $m_1 \prec m_2$ or $m_1 = m_2$.

Definition 13 (Leading, reducible, and irreducible terms). For a polynomial $P = \sum_i t_i$, the *leading term* $LT(P)$ of P is the largest term t_i according to $\prec$. Let I be an ideal over the polynomial ring $\mathbb{F}[x, y, z, \ldots]/\langle x^2 - x, y^2 - y, z^2 - z, \ldots\rangle$. We say that a term t is *reducible modulo I* if there exists a polynomial $Q \in I$ such that $t = LT(Q)$ and that t is *irreducible modulo I* otherwise.

The following fact is not hard to verify.

Fact 14. *Let I be an ideal over $\mathbb{F}[x, y, z, \ldots]/\langle x^2 - x, y^2 - y, z^2 - z, \ldots\rangle$. Then any multilinear polynomial $P \in \mathbb{F}[x, y, z, \ldots]/\langle x^2 - x, y^2 - y, z^2 - z, \ldots\rangle$ can be written uniquely as a sum $Q + R$, where $Q \in I$ and R is a linear combination of irreducible terms modulo I.*

This is what allows us to reduce polynomials modulo an ideal in a well-defined manner.

Definition 15 (Reduction operator). Let P be any multilinear polynomial in the ring $\mathbb{F}[x, y, z, \ldots]/\langle x^2 - x, y^2 - y, z^2 - z, \ldots\rangle$ and let I be an ideal over the same ring. The *reduction operator* R_I is the operator that when applied to P returns the sum of irreducible terms $R_I(P) = R$ such that $P - R \in I$.

We conclude our brief algebra review by stating two observations that are more or less immediate, but are helpful enough for us to want to highlight them explicitly.

Observation 16. *For any two ideals I_1, I_2 such that $I_1 \subseteq I_2$ and any two polynomials P, P' it holds that $R_{I_2}(P \cdot R_{I_1}(P')) = R_{I_2}(PP')$.*

Proof. Let

$$P' = Q' + R' \tag{3.1}$$

for $Q' \in I_1$ and R' a linear combination of irreducible terms over I_1. Let

$$P \cdot R_{I_1}(P') = PR' = Q + R \tag{3.2}$$

for $Q \in I_2$ and R a linear combination of irreducible terms over I_2. Then

$$PP' = PQ' + PR' = PQ' + Q + R \tag{3.3}$$

where $PQ' + Q \in I_2$. By the uniqueness in Fact 14, we conclude that the equality $R_{I_2}(PP') = R = R_{I_2}(P \cdot R_{I_1}(P'))$ holds. □

Observation 17. *Suppose that the term t is irreducible modulo the ideal I and let ρ be any partial assignment of variables in $Vars(t)$ to values in $\mathbb{F}$ such that $t\restriction_\rho \neq 0$. Then $t\restriction_\rho$ is also irreducible modulo I.*

Proof. Let m_ρ be the product of all variables in t assigned by ρ and let $\alpha = m_\rho\restriction_\rho$, where by assumption we have $\alpha \neq 0$. If there is a polynomial $Q \in I$ such that $LT(Q) = t\restriction_\rho$, then $\alpha^{-1} m_\rho Q \in I$ and $LT(\alpha^{-1} m_\rho Q) = t$, contradicting that t is irreducible. □

Proof Strategy

Let us now state the lemma on which we base the proof of Theorem 11.

Lemma 18 ([Raz98]). *Let $\mathcal{F}$ be any CNF formula and $D \in \mathbb{N}^+$ be a positive integer. Suppose that there exists a linear operator R on multilinear polynomials over $Vars(\mathcal{F})$ with the following properties:*

1. *$R(1) \neq 0$.*
2. *$R(C) = 0$ for (the translations to polynomials of) all axioms $C \in \mathcal{F}$.*
3. *For every term t with $Deg(t) < D$ and every variable x it holds that $R(xt) = R(xR(t))$.*

Then any polynomial calculus refutation of $\mathcal{F}$ (and hence any PCR refutation of $\mathcal{F}$) requires degree strictly greater than D.

The proof of Lemma 18 is not hard. The basic idea is that R will map all axioms to 0 by property 2, and further derivation steps in degree at most D will yield polynomials that also map to 0 by property 3 and the linearity of R. But then property 1 implies that no derivation in degree at most D can reach contradiction.

To prove Theorem 11, we construct a linear operator $R_{\mathcal{G}}$ that satisfies the conditions of Lemma 18 when the $(\mathcal{U}, \mathcal{V})_E$-graph $\mathcal{G}$ is an expander. First, let us describe how we make the connection between polynomials and the given $(\mathcal{U}, \mathcal{V})_E$-graph. We remark that in the rest of this section we will identify a clause C with its polynomial translation and will refer to C as a (polynomial) axiom.

Definition 19 (Term and polynomial neighbourhood). The *neighbourhood* $N(t)$ of a term t with respect to $(\mathcal{U}, \mathcal{V})_E$ is $N(t) = \{V \in \mathcal{V} \mid \mathit{Vars}(t) \cap V \neq \emptyset\}$, i.e., the family of all variable sets containing variables mentioned by t. The neighbourhood of a polynomial $P = \sum_i t_i$ is $N(P) = \bigcup_i N(t_i)$, i.e., the union of the neighbourhoods of all terms in P.

To every polynomial we can now assign a family of variable sets $\mathcal{V}'$. But we are interested in the axioms that are needed in order to produce that polynomial. That is, given a family of variable sets $\mathcal{V}'$, we would like to identify the largest set of axioms $\mathcal{U}'$ that could possibly have been used in a derivation that yielded polynomials P with $\mathit{Vars}(P) \subseteq \bigcup_{V \in \mathcal{V}'} V$. This is the intuition behind the next definition.[5]

Definition 20 (Polynomial support). For a given $(\mathcal{U}, \mathcal{V})_E$-graph and a family of variable sets $\mathcal{V}' \subseteq \mathcal{V}$, we say that a subset $\mathcal{U}' \subseteq \mathcal{U}$ is $(s, \mathcal{V}')$-*contained* if $|\mathcal{U}'| \leq s$ and $\partial_E(\mathcal{U}') \subseteq \mathcal{V}'$.

We define the *polynomial s-support $\mathit{Sup}_s(\mathcal{V}')$ of $\mathcal{V}'$ with respect to* $(\mathcal{U}, \mathcal{V})_E$, or just *$s$-support of $\mathcal{V}'$* for brevity, to be the union of all $(s, \mathcal{V}')$-contained subsets $\mathcal{U}' \subseteq \mathcal{U}$, and the s-support $\mathit{Sup}_s(t)$ of a term t is defined to be the s-support of $N(t)$.

We will usually just speak about "support" below without further qualifying this term, since the $(\mathcal{U}, \mathcal{V})_E$-graph $\mathcal{G}$ will be clear from context. The next observation follows immediately from Definition 20.

Observation 21. *Support is monotone in the sense that if $t \subseteq t'$ are two terms, then it holds that $\mathit{Sup}_s(t) \subseteq \mathit{Sup}_s(t')$.*

Once we have identified the axioms that are potentially involved in deriving P, we define the linear operator $R_{\mathcal{G}}$ as the reduction modulo the ideal generated by these axioms as in Definition 15. We will show that under the assumptions in Theorem 11 it holds that this operator satisfies the conditions in Lemma 18. Let us

[5] We remark that Definition 20 is a slight modification of the original definition of support in [AR03] that was proposed by Yuval Filmus [Fil14].

first introduce some notation for the set of all polynomials that can be generated from some axioms $\mathcal{U}' \subseteq \mathcal{U}$.

Definition 22. For a $(\mathcal{U}, \mathcal{V})_E$-graph and $\mathcal{U}' \subseteq \mathcal{U}$, we write $\mathcal{I}_E(\mathcal{U}')$ to denote the ideal generated by the polynomial axioms in $\mathcal{U}' \wedge E$.[6]

Definition 23 ($(\mathcal{U}, \mathcal{V})_E$-graph reduction). For a given $(\mathcal{U}, \mathcal{V})_E$-graph $\mathcal{G}$, the *$(\mathcal{U}, \mathcal{V})_E$-graph reduction* $R_{\mathcal{G}}$ on a term t is defined as $R_{\mathcal{G}}(t) = R_{\mathcal{I}_E(Sup_s(t))}(t)$. For a polynomial P, we define $R_{\mathcal{G}}(P)$ to be the linear extension of the operator $R_{\mathcal{G}}$ defined on terms.

Looking at Definition 23, it is not clear that we are making progress. On the one hand, we have defined $R_{\mathcal{G}}$ in terms of standard reduction operators modulo ideals, which is nice since there is a well-developed machinery for such operators. On the other hand, it is not clear how to actually compute using $R_{\mathcal{G}}$. The problem is that if we look at a polynomial $P = \sum_i t_i$ and want to compute $R_{\mathcal{G}}(P)$, then as we expand $R_{\mathcal{G}}(P) = \sum_i R_{\mathcal{G}}(t_i)$ we end up reducing terms in one and the same polynomial modulo a priori completely different ideals. How can we get any sense of what P reduces to in such a case? The answer is that if our $(\mathcal{U}, \mathcal{V})_E$-graph is a good enough expander, then this is not an issue at all. Instead, it turns out that we can pick a suitably large ideal containing the support of all the terms in P and reduce P modulo this larger ideal instead without changing anything. This key result is proven in Lemma 28 below. To establish this lemma, we need to develop a better understanding of polynomial support.

Some Properties of Polynomial Support

A crucial technical property that we will need is that if a $(\mathcal{U}, \mathcal{V})_E$-graph is a good expander in the sense of Definition 9, then for small enough sets $\mathcal{V}'$ all $(s, \mathcal{V}')$-contained subsets $\mathcal{U}' \subseteq \mathcal{U}$ as per Definition 20 are of at most half of the allowed size.

Lemma 24. *Let $(\mathcal{U}, \mathcal{V})_E$ be an (s, δ, ξ, E)-expander and let $\mathcal{V}' \subseteq \mathcal{V}$ be such that $|\mathcal{V}'| \leq \delta s/2 - \xi$. Then it holds that every $(s, \mathcal{V}')$-contained subset $\mathcal{U}' \subseteq \mathcal{U}$ is in fact $(s/2, \mathcal{V}')$-contained.*

Proof. As $|\mathcal{U}'| \leq s$ we can appeal to the expansion property of the $(\mathcal{U}, \mathcal{V})_E$-graph to derive the inequality $|\partial_E(\mathcal{U}')| \geq \delta|\mathcal{U}'| - \xi$. In the other direction, we can obtain an upper bound on the size of $\partial_E(\mathcal{U}')$ by noting that for any $(s, \mathcal{V}')$-contained set $\mathcal{U}'$ it holds that $|\partial_E(\mathcal{U}')| \leq |\mathcal{V}'|$. If we combine these bounds and use the assumption that $|\mathcal{V}'| \leq \delta s/2 - \xi$, we can conclude that $|\mathcal{U}'| \leq s/2$, which proves that $\mathcal{U}'$ is $(s/2, \mathcal{V}')$-contained. □

[6] That is, $\mathcal{I}_E(\mathcal{U}')$ is the smallest set I of multilinear polynomials that contains all axioms in $\mathcal{U}' \wedge E$ and that is closed under addition of $P_1, P_2 \in I$ and by multiplication of $P \in I$ by any multilinear polynomial over $Vars(\mathcal{U} \wedge E)$ (where as before the resulting product is implicitly multilinearized).

Even more importantly, Lemma 24 now allows us to conclude that for a small enough subset $\mathcal{V}'$ on the right-hand side of $(\mathcal{U}, \mathcal{V})_E$ it holds that in fact the whole polynomial s-support $Sup_s(\mathcal{V}')$ of $\mathcal{V}'$ on the left-hand side is $(s/2, \mathcal{V}')$-contained.

Lemma 25. *Let $(\mathcal{U}, \mathcal{V})_E$ be an (s, δ, ξ, E)-expander and let $\mathcal{V}' \subseteq \mathcal{V}$ be such that $|\mathcal{V}'| \leq \delta s/2 - \xi$. Then the s-support $Sup_s(\mathcal{V}')$ of $\mathcal{V}'$ with respect to $(\mathcal{U}, \mathcal{V})_E$ is $(s/2, \mathcal{V}')$-contained.*

Proof. We show that for any pair of $(s, \mathcal{V}')$-contained sets $\mathcal{U}_1, \mathcal{U}_2 \subseteq \mathcal{U}$ their union $\mathcal{U}_1 \cup \mathcal{U}_2$ is also $(s, \mathcal{V}')$-contained. First, by Lemma 24 we have $|\mathcal{U}_1|, |\mathcal{U}_2| \leq s/2$ and hence $|\mathcal{U}_1 \cup \mathcal{U}_2| \leq s$. Second, it holds that $\partial_E(\mathcal{U}_1), \partial_E(\mathcal{U}_2) \subseteq \mathcal{V}'$, which implies that $\partial_E(\mathcal{U}_1 \cup \mathcal{U}_2) \subseteq \mathcal{V}'$, because taking the union of two sets can only shrink the boundary. This establishes that $\mathcal{U}_1 \cup \mathcal{U}_2$ is $(s, \mathcal{V}')$-contained.

By induction on the number of $(s, \mathcal{V}')$-contained sets we can conclude that the support $Sup_s(\mathcal{V}')$ is $(s, \mathcal{V}')$-contained as well, after which one final application of Lemma 24 shows that this set is $(s/2, \mathcal{V}')$-contained. This completes the proof. □

What the next lemma says is, roughly, that if we reduce a term t modulo an ideal generated by a not too large set of polynomials containing some polynomials outside of the support of t, then we can remove all such polynomials from the generators of the ideal without changing the irreducible component of t.

Lemma 26. *Let $\mathcal{G}$ be a $(\mathcal{U}, \mathcal{V})_E$-graph and let t be any term. Suppose that $\mathcal{U}' \subseteq \mathcal{U}$ is such that $\mathcal{U}' \supseteq Sup_s(t)$ and $|\mathcal{U}'| \leq s$. Then for any term t' with $N(t') \subseteq N(Sup_s(t)) \cup N(t)$ it holds that if t' is reducible modulo $\mathcal{I}_E(\mathcal{U}')$, it is also reducible modulo $\mathcal{I}_E(Sup_s(t))$.*

Proof. If $\mathcal{U}'$ is $(s, N(t))$-contained, then by Definition 20 it holds that $\mathcal{U}' \subseteq Sup_s(t)$ and there is nothing to prove. Hence, assume $\mathcal{U}'$ is not $(s, N(t))$-contained. We claim that this implies that we can find a subformula $F \in \mathcal{U}' \setminus Sup_s(t)$ with a neighbouring subset of variables $V \in \big(\partial_E(\mathcal{U}') \cap N(F)\big) \setminus N(t')$ in the respectful boundary of $\mathcal{U}'$ but not in the neighbourhood of t'. To argue this, note that since $|\mathcal{U}'| \leq s$ it follows from Definition 20 that the reason $\mathcal{U}'$ is not $(s, N(t))$-contained is that there exist some $F \in \mathcal{U}'$ and some set of variables $V \in N(F)$ such that $V \in \partial_E(\mathcal{U}') \setminus N(t)$. Moreover, the assumption $\mathcal{U}' \supseteq Sup_s(t)$ implies that such an F cannot be in $Sup_s(t)$. Otherwise there would exist an $(s, N(t))$-contained set $\mathcal{U}^*$ such that $F \in \mathcal{U}^* \subseteq Sup_s(t) \subseteq \mathcal{U}'$, from which it would follow that $V \in \partial_E(\mathcal{U}') \cap N(\mathcal{U}^*) \subseteq \partial_E(\mathcal{U}^*) \subseteq N(t)$, contradicting $V \notin N(t)$. We have shown that $F \notin Sup_s(t) \subseteq \mathcal{U}'$ and $V \in \partial_E(\mathcal{U}') \cap N(F)$, and by combining these two facts we can also deduce that $V \notin N(Sup_s(t))$, since otherwise V could not be contained in the boundary of $\mathcal{U}'$. In particular, this means that $V \notin N(t') \subseteq N(Sup_s(t)) \cup N(t)$, which establishes the claim made above.

Fixing F and V such that $F \in \mathcal{U}' \setminus Sup_s(t)$ and $V \in \big(\partial_E(\mathcal{U}') \cap N(F)\big) \setminus N(t')$, our second claim is that if F is removed from the generators of the ideal, it still holds that if t' is reducible modulo $\mathcal{I}_E(\mathcal{U}')$, then this term is also reducible modulo $\mathcal{I}_E(\mathcal{U}' \setminus \{F\})$. Given this second claim we are done, since we can then argue by

induction over the elements in $\mathcal{U}' \setminus Sup_s(t)$ and remove them one by one to arrive at the conclusion that every term t' with $N(t') \subseteq N(Sup_s(t)) \cup N(t)$ that is reducible modulo $\mathcal{I}_E(\mathcal{U}')$ is also reducible modulo $\mathcal{I}_E(Sup_s(t))$, which is precisely what the lemma says.

We proceed to establish this second claim. The assumption that t' is reducible modulo $\mathcal{I}_E(\mathcal{U}')$ means that there exists a polynomial $P \in \mathcal{I}_E(\mathcal{U}')$ such that $t' = LT(P)$. Since P is in the ideal $\mathcal{I}_E(\mathcal{U}')$ it can be written as a polynomial combination $P = \sum_i P_i C_i$ of axioms $C_i \in \mathcal{U}' \wedge E$ for some polynomials P_i. If we could hit P with a restriction that satisfies (and hence removes) F while leaving t' and $(\mathcal{U}' \setminus \{F\}) \wedge E$ untouched, this would show that t' is the leading term of some polynomial combination of axioms in $(\mathcal{U}' \setminus \{F\}) \wedge E$. This is almost what we are going to do.

As our restriction ρ we choose any assignment with domain $\mathrm{Dom}(\rho) = V$ that E-respectfully satisfies F. Note that at least one such assignment exists since $V \in \partial_E(\mathcal{U}') \cap N(F)$ is an E-respectful neighbour of F by Definition 8. By the choice of ρ it holds that F is satisfied, i.e., that all axioms in F are set to 0. Furthermore, none of the axioms in $\mathcal{U}' \setminus \{F\}$ are affected by ρ since V is in the boundary of $\mathcal{U}'$.[7] As for axioms in E it is not necessarily true that ρ will leave all of them untouched, but by assumption ρ respects E and so any axiom in E is either satisfied (and zeroed out) by ρ or is left intact. It follows that $P\restriction_\rho$ can be be written as a polynomial combination $P\restriction_\rho = \sum_i \big(P_i\restriction_\rho\big) C_i$, where $C_i \in (\mathcal{U}' \setminus \{F\}) \wedge E$, and hence $P\restriction_\rho \in \mathcal{I}_E(\mathcal{U}' \setminus \{F\})$.

To see that t' is preserved as the leading term of $P\restriction_\rho$, note that ρ does not assign any variables in t' since $V \notin N(t')$. Hence, $t' = LT(P\restriction_\rho)$, as ρ can only make the other terms smaller with respect to $\prec$. This shows that there is a polynomial $P' = P\restriction_\rho \in \mathcal{I}_E(\mathcal{U}' \setminus \{F\})$ with $LT(P') = t'$, and hence t' is reducible modulo $\mathcal{I}_E(\mathcal{U}' \setminus \{F\})$. The lemma follows. □

We need to deal with one more detail before we can prove the key technical lemma that it is possible to reduce modulo suitably chosen larger ideals without changing the reduction operator, namely (again roughly speaking) that reducing a term modulo an ideal does not introduce any new variables outside of the generators of that ideal.

Lemma 27. *Suppose that $\mathcal{U}^* \subseteq \mathcal{U}$ for some $(\mathcal{U}, \mathcal{V})_E$-graph and let t be any term. Then it holds that $N\big(R_{\mathcal{I}_E(\mathcal{U}^*)}(t)\big) \subseteq N(\mathcal{U}^*) \cup N(t)$.*

Proof. Let $P = R_{\mathcal{I}_E(\mathcal{U}^*)}(t)$ be the polynomial obtained when reducing t modulo $\mathcal{I}_E(\mathcal{U}^*)$ and let $V \in \mathcal{V}$ be any set such that $V \notin N(\mathcal{U}^*) \cup N(t)$. We show that $V \notin N(P)$.

By the definition of $(\mathcal{U}, \mathcal{V})_E$-graphs there exists an assignment ρ to all of the variables in V that respects E. Write $t = Q + P$ with $Q \in \mathcal{I}_E(\mathcal{U}^*)$ and P a linear

[7] Recalling the remark after Definition 7, we note that we can ignore here if ρ happens to falsify axioms in $\mathcal{U} \setminus \mathcal{U}'$.

combination of irreducible monomials as in Fact 14 and apply the restriction ρ to this equality. Note that $t\restriction_\rho = t$ as V is not a neighbour of t. Moreover, $Q\restriction_\rho$ is in the ideal $\mathcal{I}_E(\mathcal{U}^*)$ because ρ does not set any variables in $\mathcal{U}^*$ and every axiom in E sharing variables with V is set to 0 by ρ. Thus, t can be written as $t = Q' + P\restriction_\rho$, with $Q' \in \mathcal{I}_E(\mathcal{U}^*)$. As all terms in P are irreducible modulo $\mathcal{I}_E(\mathcal{U}^*)$, they remain irreducible after restricting P by ρ by Observation 17. Hence, it follows that $P\restriction_\rho = P$ by the uniqueness in Fact 14 and P cannot contain any variable from V. This in turn implies that every set $V \in N(P)$ is contained in $N(\mathcal{U}^*) \cup N(t)$. □

Now we can state the formal claim that enlarging the ideal does not change the reduction operator if the enlargement is done in the right way.

Lemma 28. *Let $\mathcal{G}$ be a $(\mathcal{U}, \mathcal{V})_E$-graph and let t be any term. Suppose that $\mathcal{U}' \subseteq \mathcal{U}$ is such that $\mathcal{U}' \supseteq Sup_s(t)$ and $|\mathcal{U}'| \leq s$. Then it holds that $R_{\mathcal{I}_E(\mathcal{U}')}(t) = R_{\mathcal{I}_E(Sup_s(t))}(t)$.*

Proof. We prove that $R_{\mathcal{I}_E(\mathcal{U}')}(t) = R_{\mathcal{I}_E(Sup_s(t))}(t)$ by applying the contrapositive of Lemma 26. Recall that this lemma states that any term t' that is reducible modulo $\mathcal{I}_E(\mathcal{U}')$ and where $N(t') \subseteq N(Sup_s(t)) \cup N(t)$ is also reducible modulo $\mathcal{I}_E(Sup_s(t))$. Since every term t' in $R_{\mathcal{I}_E(Sup_s(t))}(t)$ is irreducible modulo $\mathcal{I}_E(Sup_s(t))$ and since by applying Lemma 27 with $\mathcal{U}^* = Sup_s(t)$ we have that $N(t') \subseteq N(Sup_s(t)) \cup N(t)$, it follows that t' is also irreducible modulo $\mathcal{I}_E(\mathcal{U}')$. This shows that $R_{\mathcal{I}_E(\mathcal{U}')}(t) = R_{\mathcal{I}_E(Sup_s(t))}(t)$ as claimed, and the lemma follows. □

Putting the Pieces in the Proof Together

Now we have just a couple of lemmas left before we can prove Theorem 11, which as discussed above will be established by appealing to Lemma 18.

Lemma 29. *Let $(\mathcal{U}, \mathcal{V})_E$ be an (s, δ, ξ, E)-expander with overlap $ol(\mathcal{V}) = d$. Then for any term t with $Deg(t) \leq (\delta s - 2\xi)/(2d)$ it holds that $|Sup_s(t)| \leq s/2$.*

Proof. Because of the bound on the overlap $ol(\mathcal{V})$ we have that the size of $N(t)$ is bounded by $\delta s/2 - \xi$. An application of Lemma 25 now yields the desired bound $|Sup_s(t)| \leq s/2$. □

Lemma 30. *Let $(\mathcal{U}, \mathcal{V})_E$ be an (s, δ, ξ, E)-expander with overlap $ol(\mathcal{V}) = d$. For any term t with $Deg(t) < \lfloor(\delta s - 2\xi)/(2d)\rfloor$, any term t' occurring in $R_{\mathcal{I}_E(Sup_s(t))}(t)$, and any variable x, it holds that $R_{\mathcal{I}_E(Sup_s(xt'))}(xt') = R_{\mathcal{I}_E(Sup_s(xt))}(xt')$.*

Proof. We show that $Sup_s(xt') \subseteq Sup_s(xt)$ and $|Sup_s(xt)| \leq s$, which then allows us to apply Lemma 28 and prove the lemma. To prove that $Sup_s(xt')$ is a subset of $Sup_s(xt)$, we show that $Sup_s(xt') \cup Sup_s(xt)$ is $(s, N(xt))$-contained in the sense of Definition 20. From this it follows that $Sup_s(xt') \subseteq Sup_s(xt') \cup Sup_s(xt) = Sup_s(xt)$.

Towards this goal, as $Deg(t') \leq Deg(t)$ we first observe that we can apply Lemma 29 to deduce that $|Sup_s(xt')| \leq s/2$ and $|Sup_s(xt)| \leq s/2$, and hence $|Sup_s(xt') \cup Sup_s(xt)| \leq s$, which satisfies the size condition for containment. It

remains to show that $\partial_E\big(Sup_s(xt') \cup Sup_s(xt)\big) \subseteq N(xt)$. From Lemma 27 we have that $N(t') \subseteq N(Sup_s(t)) \cup N(t)$. As $N(xt') = N(x) \cup N(t')$ and $Sup_s(t) \subseteq Sup_s(xt)$ by the monotonicity in Observation 21, it follows that

$$N(xt') = N(x) \cup N(t') \subseteq N(x) \cup N(Sup_s(t)) \cup N(t) \subseteq N(Sup_s(xt)) \cup N(xt) \ . \quad (3.4)$$

If we now consider the E-respectful boundary of the set $Sup_s(xt') \cup Sup_s(xt)$, it holds that

$$\begin{aligned}
&\partial_E \left(Sup_s(xt') \cup Sup_s(xt)\right) = \\
&\qquad = \left(\partial_E \left(Sup_s(xt')\right) \setminus N\left(Sup_s(xt)\right)\right) \cup \left(\partial_E \left(Sup_s(xt)\right) \setminus N\left(Sup_s(xt')\right)\right) \\
&\qquad \subseteq \left(N\left(xt'\right) \setminus N\left(Sup_s(xt)\right)\right) \cup \left(N\left(xt\right) \setminus N\left(Sup_s(xt')\right)\right) \\
&\qquad \subseteq N\left(xt\right) \ ,
\end{aligned} \quad (3.5)$$

where the first line follows from the boundary definition in Definition 8, the second line follows by the property of s-support that $\partial_E(Sup_s(xt)) \subseteq N(xt)$, and the last line follows from (3.4). Hence, $Sup_s(xt') \cup Sup_s(xt)$ is $(s, N(xt))$-contained.

As discussed above, we can now apply Lemma 28 to reach the desired conclusion that the equality $R_{\mathcal{I}_E(Sup_s(xt'))}(xt') = R_{\mathcal{I}_E(Sup_s(xt))}(xt')$ holds. □

Now we can prove our main technical theorem.

Proof of Theorem 11. Recall that the assumptions of the theorem are that we have a $(\mathcal{U}, \mathcal{V})_E$-graph for a CNF formula $\mathcal{F} = \bigwedge_{F \in \mathcal{U}} F \wedge E$ such that $(\mathcal{U}, \mathcal{V})_E$ is an (s, δ, ξ, E)-expander with overlap $ol(\mathcal{V}) = d$ and that furthermore for all $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$, it holds that $\bigwedge_{F \in \mathcal{U}'} F \wedge E$ is satisfiable. We want to prove that no polynomial calculus derivation from $\bigwedge_{F \in \mathcal{U}} F \wedge E = \mathcal{U} \wedge E$ of degree at most $(\delta s - 2\xi)/(2d)$ can reach contradiction.

First, if removing all axiom clauses from $\mathcal{U} \wedge E$ with degree strictly greater than $(\delta s - 2\xi)/(2d)$ produces a satisfiable formula, then the lower bound trivially holds. Otherwise, we can remove these large-degree axioms and still be left with a $(\mathcal{U}, \mathcal{V})_E$-graph that satisfies the conditions above. In order to see this, let us analyze what happens to the $(\mathcal{U}, \mathcal{V})_E$-graph if an axiom is removed from the formula.

Removing axioms from E only relaxes the conditions on respectful satisfiability while keeping all edges in the graph, so the conditions of the theorem still hold. In removing axioms from $\mathcal{U}$ we have two cases: either we remove all axioms from some subformula $F \in \mathcal{U}$ or we remove only a part of this subformula. In the former case, it is clear that we can remove the vertex F from the structure without affecting any of the conditions. In the latter case, we claim that any set $V \in \mathcal{V}$ that is an E-respectful neighbour of F remains an E-respectful neighbour of the formula F' in which large degree axioms have been removed. Clearly, the same assignments to V that satisfy F also satisfy $F' \subseteq F$. Also, V must still be a neighbour of F', for otherwise F' would not share any variables with V, which would imply that no assignment to V could satisfy F' and hence F. This would contradict the assumption

that V is an E-respectful neighbour of F. Hence, we conclude that removal of large-degree axioms can only improve the E-respectful boundary expansion of the $(\mathcal{U}, \mathcal{V})_E$-graph.

Thus, let us focus on a $(\mathcal{U}, \mathcal{V})_E$-graph $\mathcal{G}$ that has all axioms of degree at most $(\delta s - 2\xi)/(2d)$. We want to show that the operator $R_{\mathcal{G}}$ from Definition 23 satisfies the conditions of Lemma 18, from which Theorem 11 immediately follows. We can note right away that the operator $R_{\mathcal{G}}$ is linear by construction.

To prove that $R_{\mathcal{G}}(1) = R_{\mathcal{I}_E(Sup_s(1))}(1) \neq 0$, we start by observing that the size of the s-support of 1 is upper-bounded by $s/2$ according to Lemma 29. Using the assumption that for every subset $\mathcal{U}'$ of $\mathcal{U}$, $|\mathcal{U}'| \leq s$, the formula $\mathcal{U}' \wedge E$ is satisfiable, it follows that 1 is not in the ideal $\mathcal{I}_E(Sup_s(1))$ and hence $R_{\mathcal{I}_E(Sup_s(1))}(1) \neq 0$.

We next show that $R_{\mathcal{G}}(C) = 0$ for any axiom clause $C \in \mathcal{U} \wedge E$ (where we recall that we identify a clause C with its translation into a linear combination of monomials). By the preprocessing step above it holds that the degree of C is bounded by $(\delta s - 2\xi)/(2d)$, from which it follows by Lemma 29 that the size of the s-support of every term in C is bounded by $s/2$. Since C is the polynomial encoding of a clause, the leading term $LT(C)$ contains all the variables appearing in C.[8] Hence, the s-support $Sup_s(LT(C))$ of the leading term contains the s-support of every other term in C by Observation 21 and we can use Lemma 28 to conclude that $R_{\mathcal{G}}(C) = R_{\mathcal{I}_E(Sup_s(LT(C)))}(C)$. If $C \in E$, this means we are done because $\mathcal{I}_E(Sup_s(LT(C)))$ contains all of E, implying that $R_{\mathcal{G}}(C) = 0$.

For $C \in \mathcal{U}$ we cannot immediately argue that C reduces to 0, since (in contrast to [AR03]) it is not immediately clear that $Sup_s(LT(C))$ contains C. The problem here is that we might worry that C is part of some subformula $F \in \mathcal{U}$ for which the boundary $\partial_E(F)$ is not contained in $N(LT(C)) = Vars(C)$, and hence there is no obvious reason why C should be a member of any $(s, N(LT(C)))$-contained subset of $\mathcal{U}$. However, in view of Lemma 28 (applied, strictly speaking, once for every term in C) we can choose some $F \in \mathcal{U}$ such that $C \in F$ and add it to the s-support $Sup_s(LT(C))$ to obtain a set $\mathcal{U}' = Sup_s(LT(C)) \cup \{F\}$ of size $|\mathcal{U}'| \leq s/2+1 \leq s$ such that $R_{\mathcal{I}_E(Sup_s(LT(C)))}(C) = R_{\mathcal{I}_E(\mathcal{U}')}(C)$. Since $\mathcal{I}_E(\mathcal{U}')$ contains C as a generator we conclude that $R_{\mathcal{G}}(C) = R_{\mathcal{I}_E(\mathcal{U}')}(C) = 0$ also for $C \in \mathcal{U}$.[9]

It remains to prove the last property in Lemma 18 stating that $R_{\mathcal{G}}(xt) = R_{\mathcal{G}}(xR_{\mathcal{G}}(t))$ for any term t such that $Deg(t) < \lfloor(\delta s - 2\xi)/(2d)\rfloor$. We can see that

[8] We remark that this is the only place in the proof where we are using that C is (the encoding of) a clause.

[9] Actually, a slighly more careful argument reveals that C is always contained in $Sup_s(LT(C))$. This is so since for any $F \in \mathcal{U}$ with $C \in F$ it holds that any neighbours in $N(F) \setminus N(LT(C))$ have to be disrespectful, and so such an F always makes it into the support. However, the reasoning gets a bit more involved, and since we already needed to use Lemma 28 anyway we might as well apply it once more here.

this holds by studying the following sequence of equalities:

$$\begin{aligned}
R_{\mathcal{G}}(xR_{\mathcal{G}}(t)) &= \sum_{t' \in R_{\mathcal{G}}(t)} R_{\mathcal{G}}(xt') && \text{[by linearity]} \\
&= \sum_{t' \in R_{\mathcal{G}}(t)} R_{\mathcal{I}_E(Sup_s(xt'))}(xt') && \text{[by definition of } R_{\mathcal{G}}\text{]} \\
&= \sum_{t' \in R_{\mathcal{G}}(t)} R_{\mathcal{I}_E(Sup_s(xt))}(xt') && \text{[by Lemma 30]} \\
&= R_{\mathcal{I}_E(Sup_s(xt))}(xR_{\mathcal{G}}(t)) && \text{[by linearity]} \\
&= R_{\mathcal{I}_E(Sup_s(xt))}(xR_{\mathcal{I}_E(Sup_s(t))}(t)) && \text{[by definition of } R_{\mathcal{G}}\text{]} \\
&= R_{\mathcal{I}_E(Sup_s(xt))}(xt) && \text{[by Observation 16]} \\
&= R_{\mathcal{G}}(xt) && \text{[by definition of } R_{\mathcal{G}}\text{]}
\end{aligned}$$

Thus, $R_{\mathcal{G}}$ satisfies all the properties of Lemma 18, from which the theorem follows. □

Let us next show that if the slack ξ in Theorem 11 is zero, then the condition that $\mathcal{U}' \wedge E$ is satisfiable for sufficiently small $\mathcal{U}'$ is already implied by the expansion.

Lemma 31. *If a $(\mathcal{U}, \mathcal{V})_E$-graph is an $(s, \delta, 0, E)$-expander and $Vars(\mathcal{U} \wedge E) = \bigcup_{V \in \mathcal{V}} V$, then for any $\mathcal{U}' \subseteq \mathcal{U}$, $|\mathcal{U}'| \leq s$, the formula $\mathcal{U}' \wedge E$ is satisfiable.*

Proof. Let $\mathcal{U}' \subseteq \mathcal{U}$ be any subset of size at most s. First, we show that we can find a subset $\mathcal{V}' \subseteq N(\mathcal{U}')$ and an assignment ρ to the set of variables $\bigcup_{V \in \mathcal{V}'} V$ such that ρ E-respectfully satisfies $\mathcal{U}'$. We do this by induction on the number of formulas in $\mathcal{U}'$. As the $(\mathcal{U}, \mathcal{V})_E$-graph is an $(s, \delta, 0, E)$-expander it follows that $|\partial_E(\mathcal{U}')| \geq \delta|\mathcal{U}'| > 0$ for any non-empty subset $\mathcal{U}'$ and hence there exists a formula $F \in \mathcal{U}'$ and a variable set V' such that V' is an E-respectful neighbour of F and is not a neighbour of any formula in $\mathcal{U}' \setminus \{F\}$. Therefore, there is an assignment ρ to the variables in V' that E-respectfully satisfies F. By the induction hypothesis there also exists an assignment ρ' that E-respectfully satisfies $\mathcal{U}' \setminus \{F\}$ and does not assign any variables in V' as $V' \notin N(\mathcal{U}' \setminus \{F\})$. Hence, by extending the assignment ρ' to the variables in V' according to the assignment ρ, we create an assignment to the union of variables in some subset of $N(\mathcal{U}')$ that E-respectfully satisfies $\mathcal{U}'$.

We now need to show how to extend this to an assignment satisfying also E. To this end, let $\rho_{\mathcal{U}'}$ be an assignment that E-respectfully satisfies $\mathcal{U}'$ and assigns the variables in $\bigcup_{V \in \mathcal{V}'} V$ for some $\mathcal{V}' \subseteq N(\mathcal{U}')$. By another induction over the size $|\mathcal{V}'' \setminus \mathcal{V}'|$ of families $\mathcal{V}'' \supseteq \mathcal{V}'$, we show that there is an assignment $\rho_{\mathcal{V}''}$ to the variables $\bigcup_{V \in \mathcal{V}''} V$ that E-respectfully satisfies $\mathcal{U}'$ for every $\mathcal{V}' \subseteq \mathcal{V}'' \subseteq \mathcal{V}$. When $\mathcal{V}'' = \mathcal{V}'$, we just take the assignment $\rho_{\mathcal{U}'}$. We want to show that for any $V' \in \mathcal{V} \setminus \mathcal{V}''$ we can extend $\rho_{\mathcal{V}''}$ to the variables in V' so that the new assignment E-respectfully satisfies $\mathcal{U}'$. As V' respects E, there is an assignment $\rho_{V'}$ to the variables V' that satisfies all affected clauses in E. We would like to combine $\rho_{V'}$ and $\rho_{\mathcal{V}''}$ into

one assignment, but this requires some care since the intersection of the domains $V' \cap \left(\bigcup_{V \in \mathcal{V}''} V\right)$ could be non-empty. Consider therefore the subassignment $\rho^*_{V'}$ of $\rho_{V'}$ that assigns only the variables in $V' \setminus \left(\bigcup_{V \in \mathcal{V}''} V\right)$. We claim that extending $\rho_{\mathcal{V}''}$ by $\rho^*_{V'}$ creates an assignment that respects E. This is because every clause in E that has a variable in V' and was not already satisfied by $\rho_{\mathcal{V}''}$ cannot have variables in $V' \cap \left(\bigcup_{V \in \mathcal{V}''} V\right)$ (if so, $\rho_{\mathcal{V}''}$ would have been E-disrespectful) and hence every such clause must be satisfied by the subassignment $\rho^*_{V'}$.

Thus, we can find an assignment to all the variables $\cup_{V \in \mathcal{V}} V$ that E-respectfully satisfies $\mathcal{U}'$. As $\mathcal{V}$ includes all the variables in E it means that E is also fully satisfied. Hence, $\mathcal{U}' \wedge E$ is satisfiable and the lemma follows. □

This allows us to conclude this section by stating the following version of Theorem 11 for the most commonly occuring case with standard expansion without any slack.

Corollary 32. *Suppose that $(\mathcal{U}, \mathcal{V})_E$ is an $(s, \delta, 0, E)$-expander with overlap $ol(\mathcal{V}) = d$ such that $Vars(\mathcal{U} \wedge E) = \bigcup_{V \in \mathcal{V}} V$. Then any polynomial calculus refutation of the formula $\bigwedge_{F \in \mathcal{U}} F \wedge E$ requires degree strictly greater than $\delta s/(2d)$.*

Proof. This follows immediately by plugging Lemma 31 into Theorem 11. □

4 Applications

In this section, we demonstrate how to use the machinery developed in Section 3 to establish degree lower bounds for polynomial calculus. Let us warm up by reproving the bound from [AR03] for CNF formulas $\mathcal{F}$ whose clause-variable incidence graphs $G(\mathcal{F})$ are good enough expanders. We first recall the expansion concept used in [AR03] for ordinary bipartite graphs.

Definition 33 (Bipartite boundary expander). A bipartite graph $G = (U \dot\cup V, E)$ is a *bipartite (s, δ)-boundary expander* if for every set of vertices $U' \subseteq U, |U'| \leq s$, it holds that $|\partial(U')| \geq \delta |U'|$, where the *boundary* $\partial(U') = \left\{v \in V \,:\, |N(v) \cap U'| = 1\right\}$ consists of all vertices on the right-hand side V that have a unique neighbour in U' on the left-hand side.

We can simply identify the $(\mathcal{U}, \mathcal{V})_E$-graph with the standard clause-variable incidence graph $G(\mathcal{F})$ to recover the degree lower bound in [AR03] as stated next.

Theorem 34 ([AR03]). *For any CNF formula $\mathcal{F}$ and any constant $\delta > 0$ it holds that if the clause-variable incidence graph $G(\mathcal{F})$ is an (s, δ)-boundary expander, then the polynomial calculus degree required to refute $\mathcal{F}$ in polynomial calculus is $Deg(\mathcal{F} \vdash \bot) > \delta s/2$.*

Proof. To choose $G(\mathcal{F})$ as our $(\mathcal{U}, \mathcal{V})_E$-graph, we set E to be the empty formula, $\mathcal{U}$ to be the set of clauses of $\mathcal{F}$ interpreted as one-clause CNF formulas, and $\mathcal{V}$ to be the set of variables partitioned into singleton sets. As E is an empty formula every

set V respects it. Also, every neighbour of some clause $C \in \mathcal{U}$ is an E-respectful neighbour because we can set the neighbouring variable so that the clause $C \in \mathcal{U}$ is satisfied. Under this interpretation $G(\mathcal{F})$ is an $(s, \delta, 0, E)$-expander, and hence by Corollary 32 the degree of refuting $\mathcal{F}$ is greater than $\delta s/2$. □

As the second application, which is more interesting in the sense that the $(\mathcal{U}, \mathcal{V})_E$-graph is nontrivial, we show how the degree lower bound for the ordering principle formulas in [GL10] can be established using this framework. For an undirected (and in general non-bipartite) graph G, the *graph ordering principle formula* $GOP(G)$ says that there exists a totally ordered set of $|V(G)|$ elements where no element is minimal, since every element/vertex v has a neighbour $u \in N(v)$ which is smaller according to the ordering. Formally, the CNF formula $GOP(G)$ is defined over variables $x_{u,v}$, $u, v \in V(G)$, $u \neq v$, where the intended meaning of the variables is that $x_{u,v}$ is true if $u < v$ according to the ordering, and consists of the following axiom clauses:

$$\overline{x}_{u,v} \vee \overline{x}_{v,w} \vee x_{u,w} \qquad u, v, w \in V(G), u \neq v \neq w \neq u \qquad \text{(transitivity)} \qquad (4.1a)$$

$$\overline{x}_{u,v} \vee \overline{x}_{v,u} \qquad u, v \in V(G), u \neq v \qquad \text{(anti-symmetry)} \qquad (4.1b)$$

$$x_{u,v} \vee x_{v,u} \qquad u, v \in V(G), u \neq v \qquad \text{(totality)} \qquad (4.1c)$$

$$\bigvee_{u \in N(v)} x_{u,v} \qquad v \in V(G) \qquad \text{(non-minimality)} \qquad (4.1d)$$

We remark that the graph ordering principle on the complete graph K_n on n vertices is the *(linear) ordering principle formula* LOP_n (also known as a *least number principle formula*, or *graph tautology* in the literature), for which the non-minimality axioms (4.1d) have width linear in n. By instead considering graph ordering formulas for graphs G of bounded degree, one can bring the initial width of the formulas down so that the question of degree lower bounds becomes meaningful.

To prove degree lower bounds for $GOP(G)$ we need the following extension of boundary expansion to the case of non-bipartite graphs.

Definition 35 (Non-bipartite boundary expander). A graph $G = (V, E)$ is an (s, δ)-*boundary expander* if for every subset of vertices $V' \subseteq V(G)$, $|V'| \leq s$, it holds that $|\partial(V')| \geq \delta |V'|$, where the *boundary* $\partial(V') = \left\{v \in V(G) \setminus V' : \left|N(v) \cap V'\right| = 1\right\}$ is the set of all vertices in $V(G) \setminus V'$ that have a unique neighbour in V'.

We want to point out that the definition of expansion used by Galesi and Lauria in [GL10] is slightly weaker in that they do not require boundary expansion but just vertex expansion (measured as $|N(V') \setminus V'|$ for vertex sets V' with $|V'| \leq s$), and hence their result is slightly stronger than what we state below in Theorem 36. With some modifications of the definition of E-respectful boundary in $(\mathcal{U}, \mathcal{V})_E$-graphs it would be possible to match the lower bound in [GL10], but it would also make the definitions more cumbersome and so we choose not to do so here.

Theorem 36 ([GL10]). *For a non-bipartite graph G that is an (s, δ)-boundary expander it holds that $Deg(GOP(G) \vdash \bot) > \delta s/4$.*

Proof. To form the $(\mathcal{U}, \mathcal{V})_E$-graph for $GOP(G)$, we let E consist of all transitivity axioms (4.1a), anti-symmetry axioms (4.1b), and totality axioms (4.1c). The non-minimality axioms (4.1d) viewed as singleton sets form the family $\mathcal{U}$, while $\mathcal{V}$ is the family of variable sets V_v for each vertex v containing all variables that mention v, i.e., $V_v = \{x_{u,w} \mid u, w \in V(G),\ u = v \text{ or } w = v\}$.

For a vertex u, the neighbours of a non-minimality axiom $F_u = \bigvee_{v \in N(u)} x_{v,u} \in \mathcal{U}$ are variable sets V_v where v is either equal to u or a neighbour of u in G. We can prove that each $V_v \in N(F_u)$ is an E-respectful neighbour of F_u (although the particular neighbour V_u will not contribute in the proof of the lower bound). If $v \neq u$, then setting all the variables $x_{v,w} \in V_v$ to true and all the variables $x_{w,v} \in V_v$ to false (i.e., making v into the minimal element of the set) satisfies F_u as well as all the affected axioms in E. If $v = u$, we can use a complementary assignment to the one above (i.e., making $v = u$ into the maximal element of the set) to E-respectfully satisfy F_u. Observe that this also shows that all $V_v \in \mathcal{V}$ respect E as required by Definition 7.

By the analysis above, it holds that the boundary $\partial(V')$ of some vertex set V' in G yields the E-respectful boundary $\partial_E\big(\bigcup_{u \in V'} F_u\big) \supseteq \{V_v \mid v \in \partial(V')\}$ in $(\mathcal{U}, \mathcal{V})_E$. Thus, the expansion parameters for $(\mathcal{U}, \mathcal{V})_E$ are the same as those for G and we can conclude that $(\mathcal{U}, \mathcal{V})_E$ is an $(s, \delta, 0, E)$-expander.

Finally, we note that while $\mathcal{V}$ is *not* a partition of the variables of $GOP(G)$, the overlap is only $ol(\mathcal{V}) = 2$ since every variable $x_{u,v}$ occurs in exactly two sets V_u and V_v in $\mathcal{V}$. Hence, by Corollary 32 the degree of refuting $GOP(G)$ is greater than $\delta s/4$. □

With the previous theorem in hand, we can prove (a version of) the main result in [GL10], namely that there exists a family of 5-CNF formulas witnessing that the lower bound on size in terms of degree in Theorem 2 is essentially optimal. That is, there are formulas over N variables that can be refuted in polynomial calculus (in fact, in resolution) in size polynomial in N but require degree $\Omega\big(\sqrt{N}\big)$. This follows by plugging expanders with suitable parameters into Theorem 36. By standard calculations (see, for example, [HLW06]) one can show that there exist constants $\gamma, \delta > 0$ such that randomly sampled graphs on n vertices with degree at most 5 are $(\gamma n, \delta)$-boundary expanders in the sense of Definition 35 with high probability. By Theorem 36, graph ordering principle formulas on such graphs yield 5-CNF formulas over $\Theta\big(n^2\big)$ variables that require degree $\Omega(n)$. Since these formulas have polynomial calculus refutations in size $\mathrm{O}\big(n^3\big)$ (just mimicking the resolution refutations constructed in [Stå96]), this shows that the bound in Theorem 2 is essentially tight. The difference between this bound and [GL10] is that since a weaker form of expansion is required in [GL10] it is possible to use 3-regular graphs, yielding families of 3-CNF formulas.

Let us now turn our attention back to bipartite graphs and consider different flavours of pigeonhole principle formulas. We will focus on formulas over bounded-degree bipartite graphs, where we will convert standard bipartite boundary expansion as in Definition 33 into respectful boundary expansion as in Definition 9. For a bipartite graph $G = (U \dot\cup V, E)$ the axioms appearing in the different versions of the graph pigeonhole principle formulas are as follows:

$$\bigvee_{v \in N(u)} x_{u,v} \qquad u \in U \qquad \text{(pigeon axioms)} \tag{4.2a}$$

$$\overline{x}_{u,v} \vee \overline{x}_{u',v} \qquad v \in V,\, u, u' \in N(v),\, u \neq u', \qquad \text{(hole axioms)} \tag{4.2b}$$

$$\overline{x}_{u,v} \vee \overline{x}_{u,v'} \qquad u \in U,\, v, v' \in N(u),\, v \neq v' \qquad \text{(functionality axioms)} \tag{4.2c}$$

$$\bigvee_{u \in N(v)} x_{u,v} \qquad v \in V \qquad \text{(onto axioms)} \tag{4.2d}$$

The "plain vanilla" *graph pigeonhole principle formula* $PHP(G)$ is the CNF formula over variables $\{x_{u,v} \mid (u,v) \in E\}$ consisting of clauses (4.2a) and (4.2b); the *graph functional pigeonhole principle formula* $FPHP(G)$ contains the clauses of $PHP(G)$ and in addition clauses (4.2c); the *graph onto pigeonhole principle formula* $Onto\text{-}PHP_G$ contains $PHP(G)$ plus clauses (4.2d); and the *graph onto functional pigeonhole principle formula* $Onto\text{-}FPHP_G$ consists of all the clauses (4.2a)–(4.2d).

We obtain the standard versions of the PHP formulas by considering graph formulas as above over the complete bipartite graph $K_{n+1,n}$. In the opposite direction, for any bipartite graph G with $n+1$ vertices on the left and n vertices on the right we can hit any version of the pigeonhole principle formula over $K_{n+1,n}$ with the restriction ρ_G setting $x_{u,v}$ to false for all $(u,v) \notin E(G)$ to recover the corresponding graph pigeonhole principle formula over G. When doing so, we will use the observation from Section 2 that restricting a formula can only decrease the size and degree required to refute it.

As mentioned in Section 1, it was established already in [AR03] that good bipartite boundary expanders G yield formulas $PHP(G)$ that require large polynomial calculus degree to refute. We can reprove this result in our language—and, in fact, observe that the lower bound in [AR03] works also for the onto version $Onto\text{-}PHP_G$—by constructing an appropriate $(\mathcal{U}, \mathcal{V})_E$-graph. In addition, we can generalize the result in [AR03] slightly by allowing some additive slack $\xi > 0$ in the expansion in Theorem 11. This works as long as we have the guarantee that no too small subformulas are unsatisfiable.

Theorem 37. *Suppose that $G = (U \dot\cup V, E)$ is a bipartite graph with $|U| = n$ and $|V| = n - 1$ and that $\delta > 0$ is a constant such that*

- *for every set $U' \subseteq U$ of size $|U'| \leq s$ there is a matching of U' into V, and*
- *for every set $U' \subseteq U$ of size $|U'| \leq s$ it holds that $|\partial(U')| \geq \delta |U'| - \xi$.*

Then $Deg(Onto\text{-}PHP_G \vdash \bot) > \delta s/2 - \xi$.

Proof sketch. The $(\mathcal{U}, \mathcal{V})_E$-graph for $PHP(G)$ is formed by taking $\mathcal{U}$ to be the set of pigeon axioms (4.2a), E to consist of the hole axioms (4.2b) and onto axioms (4.2d), and $\mathcal{V}$ to be the collection of variable sets $V_v = \{x_{u,v} \mid u \in N(v)\}$ partitioned with respect to the holes $v \in V$. It is straightforward to check that this $(\mathcal{U}, \mathcal{V})_E$-graph is isomorphic to the graph G and that all neighbours in $(\mathcal{U}, \mathcal{V})_E$ are E-respectful (for $\bigvee_{v \in N(u)} x_{u,v} \in \mathcal{U}$ and V_v for some $v \in N(u)$, apply the partial assignment sending pigeon u to hole v and ruling out all other pigeons in $N(v) \setminus \{u\}$ for v). Moreover, using the existence of matchings for all sets of pigeons U' of size $|U'| \leq s$ we can prove that every subformula $\mathcal{U}' \wedge E$ is satisfiable as long as $|\mathcal{U}'| \leq s$. Hence, we can apply Theorem 11 to derive the claimed bound. We refer to the upcoming full-length version of [MN14] for the omitted details. □

Theorem 37 is the only place in this paper where we use non-zero slack for the expansion. The reason that we need slack is so that we can establish lower bounds for another type of formulas, namely the subset cardinality formulas studied in [Spe10, VS10, MN14]. A brief (and somewhat informal) description of these formulas is as follows. We start with a 4-regular bipartite graph to which we add an extra edge between two non-connected vertices. We then write down clauses stating that each degree-4 vertex on the left has at least 2 of its edges set to true, while the single degree-5 vertex has a strict majority of 3 incident edges set to true. On the right-hand side of the graph we encode the opposite, namely that all vertices with degree 4 have at least 2 of its edges set to false, while the vertex with degree 5 has at least 3 edges set to false. A simple counting argument yields that the CNF formula consisting of these clauses must be unsatisfiable. Formally, we have the following definition (which strictly speaking is a slightly specialized case of the general construction, but again we refer to [MN14] for the details).

Definition 38 (Subset cardinality formulas [VS10, MN14]). Suppose that $G = (U \dot\cup V, E)$ is a bipartite graph that is 4-regular except that one extra edge has been added between two unconnected vertices on the left and right. Then the *subset cardinality formula* $SC(G)$ over G has variables $x_e, e \in E$, and clauses:

- $x_{e_1} \vee x_{e_2} \vee x_{e_3}$ for every triple e_1, e_2, e_3 of edges incident to any $u \in U$,
- $\overline{x}_{e_1} \vee \overline{x}_{e_2} \vee \overline{x}_{e_3}$ for every triple e_1, e_2, e_3 of edges incident to any $v \in V$.

To prove lower bounds on refutation degree for these formulas we use the standard notion of vertex expansion on bipartite graphs, where all neighbours on the left are counted and not just unique neighbours as in Definition 33.

Definition 39 (Bipartite expander). A bipartite graph $G = (U \dot\cup V, E)$ is a *bipartite* (s, δ)*-expander* if for each vertex set $U' \subseteq U, |U'| \leq s$, it holds that $|N(U')| \geq \delta |U'|$.

The existence of such expanders with appropriate parameters can again be established by straightforward calculations (as in, for instance, [HLW06]).

Theorem 40 ([MN14]). *Suppose that $G = (U \dot\cup V, E)$ is a 4-regular bipartite $\big(\gamma n, \frac{5}{2} + \delta\big)$-expander for $|U| = |V| = n$ and some constants $\gamma, \delta > 0$, and let G' be obtained from G by adding an arbitrary edge between two unconnected vertices in U and V. Then refuting the formula $SC(G')$ requires degree $Deg(SC(G') \vdash \bot) = \Omega(n)$, and hence size $S_{\mathcal{PCR}}(SC(G') \vdash \bot) = \exp\big(\Omega(n)\big)$.*

Proof sketch. The proof is by reducing to graph PHP formulas and applying Theorem 37 (which of course also holds with onto axioms removed). We fix some complete matching in G, which is guaranteed to exist in regular bipartite graphs, and then set all edges in the matching as well as the extra added edge to true. Now the degree-5 vertex v^* on the right has only 3 neighbours and the constraint for v^* requires all of these edges to be set to false. Hence, we set these edges to false as well which makes v^* and its clauses vanish from the formula. The restriction leaves us with n vertices on the left which require that at least 1 of the remaining 3 edges incident to them is true, while the $n - 1$ vertices on the right require that at most 1 out of their incident edges is true. That is, we have restricted our subset cardinality formula to obtain a graph PHP formula.

As the original graph is a $(\gamma n, \frac{5}{2} + \delta)$-expander, a simple calculation can convince us that the new graph is a boundary expander where each set of vertices U' on the left with size $|U'| \leq \gamma n$ has boundary expansion $|\partial(U')| \geq 2\delta|U'| - 1$. Note the additive slack of 1 compared to the usual expansion condition, which is caused by the removal of the degree-5 vertex v^* from the right. Now we can appeal to Theorem 37 (and Theorem 2) to obtain the lower bounds claimed in the theorem. □

Let us conclude this section by presenting our new lower bounds for the functional pigeonhole principle formulas. As a first attempt, we could try to reason as in the proof of Theorem 37 (but adding the axioms (4.2c) and removing axioms (4.2d)). The naive idea would be to modify our $(\mathcal{U}, \mathcal{V})_E$-graph slightly by substituting the functionality axioms for the onto axioms in E while keeping $\mathcal{U}$ and $\mathcal{V}$ the same. This does not work, however—although the sets $V_v \in \mathcal{V}$ are E-respectful, the only assignment that respects E is the one that sets all variables $x_{u,v} \in V_v$ to false. Thus, it is not possible to satisfy any of the pigeon axioms, meaning that there are no E-respectful neighbours in $(\mathcal{U}, \mathcal{V})_E$. In order to obtain a useful $(\mathcal{U}, \mathcal{V})_E$-graph, we instead need to redefine $\mathcal{V}$ by enlarging the variable sets V_v, using the fact that $\mathcal{V}$ is not required to be a partition. Doing so in the appropriate way yields the following theorem.

Theorem 41. *Suppose that $G = (U \dot\cup V, E)$ is a bipartite (s, δ)-boundary expander with left degree bounded by d. Then it holds that refuting $FPHP(G)$ in polynomial calculus requires degree strictly greater than $\delta s/(2d)$. It follows that if G is a bipartite $(\gamma n, \delta)$-boundary expander with constant left degree and $\gamma, \delta > 0$, then any polynomial calculus (PC or PCR) refutation of $FPHP(G)$ requires size $\exp(\Omega(n))$.*

Proof. We construct a $(\mathcal{U}, \mathcal{V})_E$-graph from $FPHP(G)$ as follows. We let the set of clauses E consist of all hole axioms (4.2b) and functionality axioms (4.2c). We

define the family $\mathcal{U}$ to consist of the pigeon axioms (4.2a) interpreted as singleton CNF formulas. For the variables we let $\mathcal{V} = \{V_v \mid v \in V\}$, where for every hole $v \in V$ the set V_v is defined by

$$V_v = \left\{ x_{u',v'} \,\middle|\, u' \in N(v) \text{ and } v' \in N(u') \right\} \ . \tag{4.3}$$

That is, to build V_v we start with the hole v on the right, consider all pigeons u' on the left that can go into this hole, and finally include in V_v for all such u' the variables $x_{u',v'}$ for all holes v' incident to u'. We want to show that $(\mathcal{U}, \mathcal{V})_E$ as defined above satisfies the conditions in Corollary 32.

Note first that every variable set V_v respects the clause set E since setting all variables in V_v to false satisfies all clauses in E mentioning variables in V_v. It is easy to see from (4.3) that when a hole v is a neighbour of a pigeon u, the variable set V_v is also a neighbour in the $(\mathcal{U}, \mathcal{V})_E$-graph of the corresponding pigeon axiom $F_u = \bigvee_{v \in N(u)} x_{u,v}$. These are the only neighbours of the pigeon axiom F_u, as each V_v contains only variables mentioning pigeons in the neighbourhood of v. In other words, G and $(\mathcal{U}, \mathcal{V})_E$ share the same neighbourhood structure.

Moreover, we claim that every neighbour V_v of F_u is an E-respectful neighbour. To see this, consider the assignment $\rho_{u,v}$ that sets $x_{u,v}$ to true and the remaining variables in V_v to false. Clearly, F_u is satisfied by $\rho_{u,v}$. All axioms in E *not* containing $x_{u,v}$ are either satisfied by $\rho_{u,v}$ or left untouched, since $\rho_{u,v}$ assigns all other variables in its domain to false. Any hole axiom $\overline{x}_{u,v} \vee \overline{x}_{u',v}$ in E that *does* contain $x_{u,v}$ is satisfied by $\rho_{u,v}$ since $x_{u',v} \in V_v$ for $u' \in N(v)$ by (4.3) and this variable is set to false by $\rho_{u,v}$. In the same way, any functionality axiom $\overline{x}_{u,v} \vee \overline{x}_{u,v'}$ containing $x_{u,v}$ is satisfied since the variable $x_{u,v'}$ is in V_v by (4.3) and is hence assigned to false. Thus, the assignment $\rho_{u,v}$ E-respectfully satisfies F_u, and so F_u and V_v are E-respectful neighbours as claimed.

Since our constructed $(\mathcal{U}, \mathcal{V})_E$-graph is isomorphic to the original graph G and all neighbour relations are respectful, the expansion parameters of G trivially carry over to respectful expansion in $(\mathcal{U}, \mathcal{V})_E$. This is just another way of saying that $(\mathcal{U}, \mathcal{V})_E$ is an $(s, \delta, 0, E)$-expander.

To finish the proof, note that the overlap of $\mathcal{V}$ is at most d. This is so since a variable $x_{u,v}$ appears in a set $V_{v'}$ only when $v' \in N(u)$. Hence, for all variables $x_{u,v}$ it holds that they appear in at most $|N(u)| \leq d$ sets in $\mathcal{V}$. Now the conclusion that any polynomial calculus refutation of $FPHP(G)$ requires degree greater than $\delta s/(2d)$ can be read off from Corollary 32. In addition, the exponential lower bound on the size of a refutation of $FPHP(G)$ when G is a $(\gamma n, \delta)$-boundary expander G with constant left degree follows by plugging the degree lower bound into Theorem 2. □

It is not hard to show (again we refer to [HLW06] for the details) that there exist bipartite graphs with left degree 3 which are $(\gamma n, \delta)$-boundary expanders for $\gamma, \delta > 0$ and hence our size lower bound for polynomial calculus refutations of $FPHP(G)$ can be applied to them. Moreover, if $|U| = n + 1$ and $|V| = n$, then we can identify some bipartite graph G that is a good expander and hit $FPHP^{n+1}_n = FPHP(K_{n+1,n})$ with a restriction ρ_G setting $x_{u,v}$ to false for all $(u, v) \notin E$ to obtain

$FPHP_n^{n+1}\restriction_{\rho_G} = FPHP(G)$. Since restrictions can only decrease refutation size, it follows that size lower bounds for $FPHP(G)$ apply also to $FPHP_n^{n+1}$, yielding the second lower bound claimed in Section 1.

Theorem 42. *Any polynomial calculus or polynomial calculus resolution refutation of (the standard CNF encoding of) the functional pigeonhole principle* $FPHP_n^{n+1}$ *requires size* $\exp(\Omega(n))$.

5 Concluding Remarks

In this work, we extend the method developed by Alekhnovich and Razborov [AR03] for proving degree lower bounds on refutations of CNF formulas in polynomial calculus. Instead of looking at the clause-variable incidence graph $G(F)$ of the formula F as in [AR03], we allow clustering of clauses and variables and reason in terms of the incidence graph G' defined on these clusters. We show that the CNF formula F requires high degree to be refuted in polynomial calculus whenever this clustering can be done in a way that "respects the structure" of the formula and so that the resulting graph G' has certain expansion properties.

This provides us with a unified framework within which we can reprove previously established degree lower bounds in [AR03, GL10, MN14]. More importantly, this also allows us to obtain a degree lower bound on the functional pigeonhole principle defined on expander graphs, solving an open problem from [Raz02]. It immediately follows from this that the (standard CNF encodings of) the usual functional pigeonhole principle formulas require exponential proof size in polynomial calculus resolution, resolving a question on Razborov's problems list [Raz15] which had (quite annoyingly) remained open. This means that we now have an essentially complete understanding of how the different variants of pigeonhole principle formulas behave with respect to polynomial calculus in the standard setting with $n+1$ pigeons and n holes. Namely, while Onto-FPHP formulas are easy, both FPHP formulas and Onto-PHP formulas are exponentially hard in n even when restricted to bounded-degree expanders.

A natural next step would be to see if this generalized framework can also be used to attack other interesting formula families which are known to be hard for resolution but for which there are currently no lower bounds in polynomial calculus. In particular, can our framework or some modification of it prove a lower bound for refuting the formulas encoding that a graph does not contain an independent set of size k, which were proven hard for resolution in [BIS07]? Or what about the formulas stating that a graph is k-colorable, for which resolution lower bounds were established in [BCMM05]?

Returning to the pigeonhole principle, we now understand how different encodings behave in polynomial calculus when we have $n+1$ pigeons and n holes. But what happens when we increase the number of pigeons? For instance, do the formulas become easier if we have n^2 pigeons and n holes? (This is the point where lower bound techniques based on degree break down.) What about arbitrary many

pigeons? In resolution these questions are fairly well understood, as witnessed by the works of Raz [Raz04a] and Razborov [Raz01, Raz03, Raz04b], but as far as we are aware they remain wide open for polynomial calculus.

Finally, we want to point out an intriguing contrast between our work and that of Alekhnovich and Razborov. As discussed in the introduction, the main technical result in [AR03] is that when the incidence graph of a set of polynomial equations is expanding and the polynomials are immune, i.e., have no low-degree consequences, then refuting this set of equations is hard with respect to polynomial calculus degree. Since clauses of width w have maximal immunity w, it follows that for a CNF formula F expansion of the clause-variable incidence graph $G(F)$ is enough to imply hardness. A natural way of interpreting our work would be to say that we simply extend this result to a slightly more general constraint-variable incidence graph. On closer inspection, however, this analogy seems to be misleading, and since we were quite surprised by this ourselves we want to elaborate briefly on this.

For the functional pigeonhole principle, the pigeon and functional axioms for a pigeon u taken together imply the polynomial equation $\sum_{v \in N(u)} x_{u,v} = 1$ (summing over all holes $v \in N(u)$ to which the pigeon u can fly). Since this is a degree-1 consequence, it shows that the pigeonhole axioms in FPHP formulas have *lowest possible immunity* modulo the set E consisting of hole and functionality axioms. Nevertheless, our lower bound proof still works, and only needs expansion of the constraint-variable graph although the immunity of the constraints is non-existent.

On the other hand, the constraint-variable incidence graph of a random set of parity constraints is expanding asymptotically almost surely, and since over fields of characteristic distinct from 2 parity constraints have high immunity (see, for instance, [Gre00]), the techniques in [AR03] can be used to prove strong degree lower bounds in such a setting. However, it seems that our framework of respectful boundary expansion is inherently unable to establish this result. The problem is that (as discussed in the footnote after Definition 9) it is not possible to group variables together in such a way as to ensure respectful neighbourhood relations. At a high level, it seems that the main ingredient needed for our technique to work is that clauses/polynomials and variables can be grouped together in such a way that the effects of assignments to a group of variables can always be contained in a small neighbourhood of clauses/polynomials, which the assignments (mostly) satisify, and do not propagate beyond this neighbourhood. Functional pigeonhole principle formulas over bounded-degree graphs have this property, since assigning a pigeon u to a hole v only affects the neighbouring holes of u and the neighbouring pigeons of v, respectively. There is no such way to contain the effects locally when one starts satisfying individual equations in an expanding set of parity constraints, however, regardless of the characteristic of the underlying field.

In view of this, it seems that our techniques and those of [AR03] are closer to being orthogonal rather than parallel. It would be desirable to gain a deeper understanding of what is going on here. In particular, in comparison to [AR03], which gives clear, explicit criteria for hardness (is the graph expanding? are the

polynomials immune?), our work is less explicit in that it says that hardness is implied by the existence of a "clustered clause-variable incidence graph" with the right properties, but gives no guidance as to if and how such a graph might be built. It would be very interesting to find more general criteria of hardness that could capture both our approach and that of [AR03], and ideally provide a unified view of these lower bound techniques.

Acknowledgements

We are grateful to Ilario Bonacina, Yuval Filmus, Nicola Galesi, Massimo Lauria, Alexander Razborov, and Marc Vinyals for numerous discussions on proof complexity in general and polynomial calculus degree lower bounds in particular. We want to give a special thanks to Massimo Lauria for several insightful comments on an earlier version of this work, which allowed us to simplify the construction (and improve the parameters in the results) considerably, and to Alexander Razborov for valuable remarks on a preliminary version of this manuscript that, in particular, helped to shed light on the similarities with and differences from the techniques in [AR03]. We also want to acknowledge useful discussions with the participants of the Dagstuhl workshop 15171 *Theory and Practice of SAT Solving* in April 2015. Finally, we are most thankful for the very detailed feedback provided by the anonymous reviewers, which helped improve this manuscript considerably.

Bibliography

[ABRW02] 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*.

[AR03] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. *Proceedings of the Steklov Institute of Mathematics*, 242:18–35, 2003. Available at `http://people.cs.uchicago.edu/~razborov/files/misha.pdf`. Preliminary version in *FOCS '01*.

[BCMM05] Paul Beame, Joseph C. Culberson, David G. Mitchell, and Cristopher Moore. The resolution complexity of random graph k-colorability. *Discrete Applied Mathematics*, 153(1-3):25–47, December 2005.

[BGIP01] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *Journal of Computer and System Sciences*, 62(2):267–289, March 2001. Preliminary version in *CCC '99*.

[BIS07] Paul Beame, Russell Impagliazzo, and Ashish Sabharwal. The resolution complexity of independent sets and vertex covers in random graphs. *Computational Complexity*, 16(3):245–297, October 2007.

[Bla37] Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version in *STOC '99*.

[CEI96] 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)*, pages 174–183, May 1996.

[CR79] Stephen A. Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979.

[CS88] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[Fil14] Yuval Filmus. On the Alekhnovich–Razborov degree lower bound. Manuscript. Available at `http://www.cs.toronto.edu/~yuvalf/AlRa.pdf`, October 2014.

[GL10] Nicola Galesi and Massimo Lauria. Optimality of size-degree trade-offs for polynomial calculus. *ACM Transactions on Computational Logic*, 12:4:1–4:22, November 2010.

[Gre00] Frederic Green. A complex-number Fourier technique for lower bounds on the mod-m degree. *Computational Complexity*, 9(1):16–38, January 2000.

[Gri98] Dima Grigoriev. Tseitin's tautologies and lower bounds for Nullstellensatz proofs. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS '98)*, pages 648–652, November 1998.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, October 2006.

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[MN14] Mladen Mikša and Jakob Nordström. Long proofs of (seemingly) simple formulas. In *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *Lecture Notes in Computer Science*, pages 121–137. Springer, July 2014.

[MN15] Mladen Mikša and Jakob Nordström. A generalized method for proving polynomial calculus degree lower bounds. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 467–487, June 2015.

[Nor13] Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

[Raz98] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, December 1998.

[Raz01] Alexander A. Razborov. Improved resolution lower bounds for the weak pigeonhole principle. Technical Report TR01-055, Electronic Colloquium on Computational Complexity (ECCC), July 2001.

[Raz02] Alexander A. Razborov. Proof complexity of pigeonhole principles. In *5th International Conference on Developments in Language Theory, (DLT '01), Revised Papers*, volume 2295 of *Lecture Notes in Computer Science*, pages 100–116. Springer, July 2002.

[Raz03] Alexander A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theoretical Computer Science*, 1(303):233–243, June 2003.

[Raz04a] Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *Journal of the ACM*, 51(2):115–138, March 2004. Preliminary version in *STOC '02*.

[Raz04b] Alexander A. Razborov. Resolution lower bounds for perfect matching principles. *Journal of Computer and System Sciences*, 69(1):3–27, August 2004. Preliminary version in *CCC '02*.

[Raz15] Alexander A. Razborov. Possible research directions. List of open problems (in proof complexity and other areas) available at `http://people.cs.uchicago.edu/~razborov/teaching/`, 2015.

[Rii93] Søren Riis. *Independence in Bounded Arithmetic*. PhD thesis, University of Oxford, 1993.

[Spe10] Ivor Spence. sgen1: A generator of small but difficult satisfiability benchmarks. *Journal of Experimental Algorithmics*, 15:1.2:1.1–1.2:1.15, March 2010.

[Stå96] Gunnar Stålmarck. Short resolution proofs for a sequence of tricky formulas. *Acta Informatica*, 33(3):277–280, May 1996.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

[VS10] Allen Van Gelder and Ivor Spence. Zero-one designs produce small hard SAT instances. In *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT '10)*, volume 6175 of *Lecture Notes in Computer Science*, pages 388–397. Springer, July 2010.