Some Trade-off Results for Polynomial Calculus

Chris Beck
Princeton University
cbeck@princeton.edu

Jakob Nordström
KTH Royal Institute of Technology
jakobn@kth.se

Bangsheng Tang
Tsinghua University
bangsheng.tang@gmail.com

April 6, 2012

Abstract

Although the satisfiability problem is believed to be intractable in the worst case, today state-of-the-art SAT solvers are routinely used to solve real-world instances with up to millions of variables. The two main bottlenecks for such solvers are the amounts of time and memory used. Generally speaking, the currently best solvers use conflict-driven clause learning (CDCL), whereas algebraic approaches based on Gröbner bases, although theoretically promising, have so far failed to deliver comparable performance.

In the field of proof complexity, time and memory resources correspond to the size and space of proofs. CDCL solvers are known to search for proofs in *resolution*, and a long line of work has investigated the relations between size and space in this proof system, culminating in the recent papers [Ben-Sasson and Nordström 2011, Beame, Beck, and Impagliazzo 2012]. *Polynomial calculus (PC)*, which is the system corresponding to solvers using Gröbner bases, has remained very much less understood from the perspective of space, however, and because of the importance of memory usage in actual solvers, shedding light on this has been an interesting theoretical goal.

In this paper, we extend essentially all known trade-off results for resolution to polynomial calculus and the stronger proof system *polynomial calculus resolution (PCR)*, which unifies resolution and PC. Our collection of trade-offs cover a wide range of values for the space complexity of formulas, from constant all the way up to superlinear, and most of the trade-offs are superpolynomial or even exponential. Interestingly, we obtain our trade-offs for the same formulas as in [Ben-Sasson and Nordström 2011; Beame, Beck and Impagliazzo 2012], which in particular means that the upper bounds hold for resolution while the lower bounds hold for PCR. This shows that there are formulas for which adding algebraic reasoning on top of resolution does not improve the time-space trade-off properties in any significant way. As a byproduct of our analysis, we are also able to prove trade-offs between space and degree in PCR and PC exactly matching analogous results for space versus width in resolution.

In terms of techniques, we combine ingredients from several previous papers, such as the notion of degree as a key resource for PC-proofs in [Impagliazzo, Pudlák, and Sgall 1999], subadditive complexity measures of progress for proofs in [Ben-Sasson and Wigderson 2001], hardness amplification by substitution in [Ben-Sasson and Nordström 2011], the random restrictions and isoperimetric inequalities in [Beame, Beck, and Impagliazzo 2012], and polynomial calculus over binomial ideals in [Buss, Grigoriev, Impagliazzo, and Pitassi 2001]. Studying so-called pebbling formulas and Tseitin formulas, and applying the right mix of the above constituents, yields our results.

1 Introduction

Ever since Cook's landmark paper [Coo71], the question of how hard it is to prove formulas in propositional logic has been a central problem in theoretical computer science. While it is nowadays generally believed that this problem should be intractable in the worst case, there have nevertheless been impressive algorithmic developments in the last 10–15 years in this area, which is usually referred to as SAT solving. Today SAT solvers are routinely deployed in a number of different application areas to solve real-world problem instances with hundreds of thousands or even millions of variables.

A somewhat surprising aspect of this development is that at the core, a clear majority of the state-of-the-art SAT solvers today are still based on the fairly simple Davis-Putnam-Logemann-Loveland (DPLL) procedure [DLL62, DP60] from the early 1960s augmented with clause learning [BS97, MS96]; these programs are also known as conflict-driven clause learning (CDCL) solvers. Despite the fact that such SAT solvers in effect search for proofs in the relatively weak proof system *resolution* [Bla37], for which numerous exponential lower bounds are known, CDCL solvers have dominated the international SAT competition [SAT] in recent years. Another approach is to do algebraic SAT solvers based on Gröbner basis computations, which correspond to the proof system *polynomial calculus* [CEI96]. Intriguingly, although theoretical results seem to hold out the promise that such algebraic SAT solvers could be far better than those based on resolution, this promise has so far failed to materialize. Looking at state-of-the-art algebraic solvers such as PolyBoRi [BD09], in general they seem to be an order of magnitude slower than the best CDCL solvers.[1] In a nutshell, the problem is that although polynomial calculus seems to be stronger as a method of reasoning than resolution, we do not yet know how to do *automated* reasoning efficiently in polynomial calculus whereas DPLL plus clause learning turns out to be a very efficient proof search procedure for resolution.

The field of proof complexity, as initiated by Cook and Reckhow [CR79], also studies the hardness of proving propositional logic formulas, but from the slightly different (and non-constructive) angle of how succinct such proofs can be, regardless of how hard or easy it is to find these proofs. In its most general form, a proof system for a language L is defined as a predicate $\mathcal{P}(x, \pi)$, computable in time polynomial in the total size of the inputs $|x| + |\pi|$, which has the property that for all $x \in L$ there is a string π (a *proof*) such that $\mathcal{P}$ accepts the input (x, π), whereas for any $x \notin L$ it holds for all π that $\mathcal{P}$ rejects (x, π). A proof system is said to be polynomially bounded if for every $x \in L$ there is a proof π_x of size at most polynomial in $|x|$. A *propositional proof system* is a proof system for the language of tautologies in propositional logic.

An important motivation for proof complexity has been the problem of P vs. NP, the connection being that [CR79] showed that co-NP $\neq$ NP, if and only if there are no polynomially bounded propositional proof systems. Cook's program refers to a long line of work towards understanding this question by studying the simplest proof systems first and proving superpolynomial lower bounds, with the goal of uncovering more and more general techniques to resolve the question. This goal remains very distant, however, and today a good deal of research in the area is primarily driven by other concerns.

One such other concern is the connection to practical SAT solving. By studying proof systems that are, or could potentially be, used by SAT solvers, one can hope to gain a better understanding of the potential and limitations of such solvers. There is a growing literature of such papers, with [AFT11, BJ10, PD11] being a very subjective pick of just recent interesting examples studying resolution and CDCL solvers.

The two main bottlenecks for modern SAT solvers are running time and memory usage. By studying proof size/length, proof space, and trade-offs between these two measures in different proof systems, we want to understand how the important resources of time and space are connected to one another and whether they can be optimized simultaneously or have to be traded for one another in SAT solvers using these proof systems.[2] We want to point out here that this is not necessarily just a theoretical exercise—for instance,

[1] For completeness, it should be mentioned that the paper [BDG+09] reports that PolyBoRi can be faster than CDCL solvers on certain specific industrial instances, but this does not change the overall picture.

[2] In fact, because we think of proof size as corresponding to running time (albeit for a nondeterminyistic program), trade-off

recent experimental work reported in [JMNŽ12] seems to suggest that space complexity in resolution is correlated with hardness in practice for CDCL solvers.

An additional motivation for our work coming from a computational complexity standpoint, but distinct from Cook's program, is a recent conjecture in the paper [ACL+12] of the third author and co-authors. They study tree-width parameterized SAT[3] and show that across several ranges of the parameter it is complete for natural complexity classes lying between L and P. This implies that certain natural longstanding complexity conjectures, such as $\mathsf{SAC}^1 \not\subseteq \mathsf{SC}$,[4] can be rephrased as natural questions about time-space trade-offs for SAT instances of small tree-width. A recent result [BBI12] by the first author and co-authors demonstrated instances across a wide range of the tree-width parameter for which a quantitatively weaker form of this conjecture is true for the restricted class of algorithms corresponding to resolution. This connection was somewhat unexpected, and motivates the program of extending these results to stronger proof systems and improving the results quantitatively as a new way to give evidence for such complexity conjectures.

Concluding this brief general discussion, let us mention that some good starting points for a further study of proof complexity in general are [Bea04, CK02, Seg07], while the upcoming survey [Nor12c] by the second author focuses specifically on time-space trade-offs and related questions. On the more practical side, a recent and very comprehensive reference on SAT solving is [BHvMW09].

1.1 Previous Work

Any formula in propositional logic can be converted to a CNF formula that is only linearly larger and is unsatisfiable if and only if the original formula is a tautology. Therefore, any sound and complete system for refuting unsatisfiable CNF formulas can be considered as a general propositional proof system. All proof systems considered in this paper are of this type.

The *resolution* proof system already mentioned above began to be investigated in connection with automated theorem proving in the 1960s [DLL62, DP60, Rob65]. In resolution, one derives new disjunctive clauses from the clauses of the original CNF formula until a contradiction is reached. For this proof system, it is most straightforward to prove bounds on the *length* of refutations, i.e., the number of clauses, rather than on size, measured as the total number of symbols in the refutations. The two measures are anyhow easily seen to be polynomially related. One of the early break-throughs in proof complexity was the result by Haken [Hak85] that CNF formulas encoding the pigeonhole principle (PHP formulas) require proofs of exponential length. There have been a sequence of follow-up papers establishing quantitatively stronger bounds for other formula families in, for instance, [BKPS02, BW01, CS88, Urq87].

The study of space in resolution was initiated by Esteban and Torán [ET01] and was later extended to a more general setting including other proof systems by Alekhnovich et al. [ABRW02]. Intuitively, the (clause) space of a resolution refutation is the maximal number of clauses one needs to keep in memory while verifying the refutation. Perhaps somewhat surprisingly, it turns out that linear space is enough to refute any formula, and a sequence of papers [ABRW02, BG03, ET01] have proven matching lower bounds.

Another sequence of papers [Nor09a, NH08, BN08] involving the second author studied the relation between length and space, showing that hardness with respect to space is "maximally uncorrelated" with hardness with respect to length. More formally, while it follows from [AD08] that small space complexity implies the existence of short resolution refutations, it was established in [BN08] that there exist explicit

results in this context are often referred to as proof complexity "time-space trade-offs." We will use the terms "size-space trade-off" and "time-space trade-off" interchangeably in this paper.

[3] We will not discuss tree-width in detail in this paper; for a discussion of tree-width parameterized SAT see [ACL+12, AR02].

[4] SC is the class of problems that can be solved in simultaneous polynomial time and polylogarithmic space on a Turing machine. An outstanding conjecture in [Coo85] is that $\mathsf{NC} \not\subseteq \mathsf{SC}$, where NC is the class of circuit of polynomial size and polylogarithmic depth. [ACL+12] demonstrated that SAT on instances of small tree-width constitutes a complete problem for the intermediate class $\mathsf{SAC}^1 = \mathsf{LOGCFL}$, and made a more general conjecture that in particular implies $\mathsf{SAC}^1 \not\subseteq \mathsf{TISP}\left(n^{O(1)}, n^{o(1)}\right)$.

formulas that are maximally easy with respect to length, having refutations in length $O(n)$, but which are hard for space in that their clause space complexity is $\Omega(n/\log n)$ (and this separation is optimal).

Regarding trade-offs between length and space, some results in restricted settings were presented in [Ben09, Nor09b] and strong trade-offs for full, unrestricted resolution were finally obtained in [BN11]. These trade-offs only apply for space smaller than the linear worst-case upper bound, however. In the even more recent work [BBI12], trade-off results that extend even to superlinear space were obtained.

Let us conclude our overview of resolution by mentioning an important auxiliary measure, namely *width*, defined as the size of a largest clause in a refutation. Ben-Sasson and Wigderson [BW01] proved that length and width are tightly connected in resolution in that a formula has a short refutation if and only if it also has a (reasonably) narrow one. Atserias and Dalmau [AD08] showed that clause space is an upper bound on width. What this means is that to prove strong lower bounds on proof length or proof space in resolution, it is sufficient to establish strong lower bounds on the width of any refutation. It was shown in [BN08], however, that small width complexity does not imply anything about the space complexity of a formula, and [Ben09] established strong trade-offs between width and space, exhibiting formulas that can be refuted in constant space and also in constant width, but where optimizing one of these measures in a refutation leads to essentially worst-case behavious for the other measure.

While the starting point of our paper is resolution and results known for this proof system, our main focus is the *polynomial calculus (PC)* proof system, which was introduced in [CEI96] under the name "Gröbner proof system." In polynomial calculus, clauses are interpreted as multilinear polynomial equations over some field, and an unsatisfiable CNF formula is refuted by showing that there is no common root for the polynomial equations corresponding to all the clauses. The size of a PC-refutation is measured as the total number of monomials in the refutation counted with repetitions, whereas length is the total number of derived polynomials, again counted with repetitions. Notice that this means that in polynomial calculus, length and size are a priori not necessarily polynomially related, and so we focus here on the size measure. The minimal size of refuting a formula in PC turns out to be closely related to the total degree of the polynomials appearing in the refutation [IPS99], and a number of strong lower bounds on proof size have been obtained by proving degree lower bounds in, for instance, [AR03, BGIP01, BI10, IPS99, Raz98].

The treatment of negated and unnegated literals in polynomial calculus is asymetric, which leads to that low-level details such as the sign of literals can have rather dramatic impact on complexity measures. To get a cleaner, more symmetric theoretical treatment, where the results do not depend on such particularities of the encoding, [ABRW02] introduced *polynomial calculus resolution (PCR)*. It is straightforward to show that PCR generalizes and unifies both resolution and polynomial calculus (hence the name).

The connection between size and degree that we just discussed is easily verified to hold for PCR as well, so the size lower bounds cited above transfer to this system. Space in PCR (and PC) is measured as the maximum number of monomials one has to remember while verifying the refutation, which is the natural analogue of clause space in resolution. [ABRW02] established nontrivial PCR space lower bound, but only for formulas of unbounded width (namely, PHP formulas). It was only very recently that [FLN+12] showed space lower bounds for formulas of constant width, and there is still quite a large gap between the best known worst-case lower and upper bounds, not only for PCR but even for PC.

As to trade-offs between size and space, we are not aware of any such results for PC or PCR except the again very recent paper [HN12]. Strictly speaking, however, it is somewhat debatable whether these results should be described as "true" trade-offs. The issue is that while [HN12] shows that any refutations in small space must have very large length, no such small-space refutations are known to exist (and indeed, the actual space complexity of the formulas looks likely to be larger than where the trade-off kicks in).

As noted above, degree is an important auxiliary measure in PC and PCR, playing a role similar to that of width in resolution. However, whereas the relationship between size and degree in PC/PCR is analogous to that between length and width in resolution, it is not known whether monomial space and degree behave with respect to each other as do clause space and width in resolution.

1.2 Our Results

As discussed above, there is a wealth of results on space lower bounds and time-space trade-offs for resolution. When it comes to PC and PCR, however, the situation has been very different to date, and as far as we are aware no "true" trade-offs (as explained above) have been known for any proof complexity measures.

In this paper, we extend the trade-offs in [Ben09, BN11, BBI12], i.e., essentially all known trade-offs for resolution,[5] to polynomial calculus and PCR. Our first result is that there is a strong trade-off between degree and monomial space in polynomial calculus and PCR, completely analogous to the trade-off between width and clause space in resolution. (We refer to Section 2 and Appendix B for precise definitions of terminology and notation used below.)

Theorem 1.1. *There is a family of 3-CNF formulas F_n of size $\Theta(n)$ that can be refuted in polynomial calculus in degree $Deg_{\mathcal{PC}}(F_n \vdash \bot) = \mathrm{O}(1)$ and also in monomial space $Sp_{\mathcal{PC}}(F_n \vdash \bot) = \mathrm{O}(1)$, but such that for any PCR-refutation $\pi_n : F_n \vdash \bot$ it holds that $Sp(\pi_n) \cdot Deg(\pi_n) = \Omega(n/\log n)$.*

What this theorem says is that although the formulas F_n can be refuted in essentially minimal degree and essentially minimal space even in PC, when we optimize one of these measures the other has to blow up to almost worst possible in PCR (the worst-case upper bound for both measures is linear in n). This result follows by studying the same so-called pebbling formulas as in [Ben09] and doing a careful analysis of the proofs in [BN11], which yields a simple but very useful generalization of the techniques there.

Our first set of time-space trade-off results follows by applying the same generalization of [BN11] to other pebbling formulas and combining this with random restrictions to obtain trade-offs where the upper bounds holds for PC (and resolution) while the lower bounds apply for the stronger PCR proof system. There is a slight loss in the parameters as compared to the results for resolution in [BN11], however, which is due to the random restriction argument, and in particular we do not get tightly matching upper and lower bounds. The trade-offs we obtain are still fairly dramatic, however, and a nice extra feature is that they also hold even if we allow the PCR-refutations to use exponentially stronger *semantic* rules where anything that follows semantically from what is currently in memory can be derived in one single step instead of by a sequence of syntactic steps. As in [BN11], the fact that we are working with pebbling formulas means that we can only get time-space trade-offs in the sublinear space regime using these techniques.

As a first example, we show that for arbitrarily small but growing space complexity, there can be super-polynomial size-space trade-offs for PC and PCR.

Theorem 1.2 (Informal). *Let $g(n) = \omega(1)$ be any arbitrarily slowly growing function[6] and fix any $\varepsilon > 0$. Then there are explicitly constructible 6-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that the following holds:*

- *The formulas F_n are refutable in polynomial calculus in total space $TotSp_{\mathcal{PC}}(F_n \vdash \bot) = \mathrm{O}(g(n))$.*
- *There are PC-refutations π_n of F_n in simultaneous size $S(\pi_n) = \mathrm{O}(n)$ and total space $TotSp(\pi_n) = \mathrm{O}\left(\left(n/g(n)^2\right)^{1/3}\right)$.*
- *Any PCR-refutation of F_n in monomial space $\mathrm{O}\left(\left(n/(g(n)^3 \log n)\right)^{1/3-\varepsilon}\right)$ must have superpolynomial size.*

Note that this trade-off is quite robust in the sense that for the whole range of space complexity from $\omega(1)$ up to almost $n^{1/3}$ the proof size required in superpolynomial. Note also that the trade-off result is

[5] It also seems likely that the trade-offs in [Nor09b] would carry over to PC/PCR, but since these results are clearly more artificial we have not looked into this.

[6] Technically speaking, we also need $g(n) = \mathrm{O}\left(n^{1/7}\right)$ here, i.e., that $g(n)$ does not grow too quickly. This restriction is inconsequential since for faster-growing functions other results presented in this paper yield stronger trade-offs anyway.

nearly tight in the sense that the superpolynomial lower bound on size in terms of space reaches up to very close to where the linear upper bound kicks in.

Let us also give an example of an exponential trade-off, where the proof size blows up from linear to $\exp n^{\varepsilon}$ when space is optimized.

Theorem 1.3 (Informal). *There is a family of explicitly constructible 6-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that the following holds:*

1. *The formulas F_n are refutable in PC in total space $TotSp_{\mathcal{PC}}(F_n \vdash \bot) = \mathrm{O}\big(n^{1/11}\big)$.*
2. *There are PC-refutations π_n of F_n in size $S(\pi_n) = \mathrm{O}(n)$ and total space $TotSp(\pi_n) = \mathrm{O}\big(n^{3/11}\big)$.*
3. *Any PCR-refutation of F_n in monomial space at most $n^{2/11}/(10 \log n)$ must have size at least $\big(n^{1/11}\big)!$.*

For the second set of time-space trade-off results, we generalize the very recent trade-offs for Tseitin formulas in [BBI12] from resolution to PCR. In resolution, these are the only known trade-off lower bounds which hold for superlinear space. Quantitatively, what they show is that for the formulas in question, if the space is reduced below a polynomial factor of the size of the smallest known proofs, the size must grow as a super constant power of the optimal size. Besides this, we give several technical improvements which allow us obtain the result for 8-CNFs and not just CNFs of unbounded width.

Theorem 1.4. *Let W be any function of n at most n^{ε} for some suitable $\varepsilon > 0$, and let $\mathbb{F}$ be a field of odd characteristic. There is an explicitly constructible family of 8-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and tree-width $W(n)$ such that the following holds:*

1. *The formulas F_n have resolution refutations π_n in (short) length $L(\pi_n) \leq n^{O(1)}2^W$ and clause space $Sp(\pi_n) \leq 2^W + n^{O(1)}$.*
2. *They also have π'_n in (small) clause space $Sp(\pi'_n) = \mathrm{O}\big(W \log n\big)$ and length $L(\pi'_n) \leq 2^{\mathrm{O}(W \log n)}$.*
3. *For any PCR-refutation π_n of F_n over $\mathbb{F}$, it holds that $S(\pi_n) = \left(\frac{2^{\Omega(W)}}{Sp(\pi_n)}\right)^{\Omega\left(\frac{\log \log n}{\log \log \log n}\right)}$.*

This result would be interesting even if $W(n)$ were a nondescript parameter, but the fact that it is the tree-width is the reason for the connection with [ACL+12]. In [ACL+12], it was shown that the resolution upper bounds mentioned above can in fact be obtained by a tree-width based algorithm with little overhead and further that a smooth tradeoff upper bound exists between the two ranges. The [ACL+12] conjecture, mentioned in the introduction, is that this algorithm cannot be improved.

Theorem 1.4 can be interpreted as evidence supporting at least a weak form of the conjecture—it places hard limits on how much a restricted class of algorithms could concievably improve over their algorithm. While an important open question in this regard is improving the exponent obtained in the lower bound argument, it is also interesting from the standpoint of the conjecture to generalize the lower bound to stronger proof systems, since this will cover a broader class of algorithms.

1.3 Outline of This Paper

The rest of this paper is organized as follows. In Section 2, we give a detailed overview of our results and describe the main technical ingredients in the proofs. Our intention is that all results claimed in this paper should be clear from this overview, and that the proof sketches should contain all salient features of the arguments. By the time we finish Section 2, however, we have already run out of space, so the rest of the paper is provided in the form of appendices.

To keep the context, Appendix A contains concluding remarks including a discussion of a few of the many fascinating problems in this area that remain open. Formal proof complexity preliminaries are given in Appendix B. In Appendix C, we do a careful study of the techniques in [BN11], and our generalization immediately allows us to derive space-degree trade-offs for polynomial calculus in Appendix D. In Appendix E, we prove time-space trade-offs for polynomial calculus in the sublinear regime. In Appendix F, we show an isoperimetric inequality for a certain kind of graphs which is instrumental for obtaining CNF formulas with strong trade-off properties, and in Appendix G we extend the time-space trade-offs for super-linear space in [BBI12] from resolution to PCR.

2 Overview of Results and High-Level Proofs

In this section, we describe which ingredients go into our results stated in Section 1.2 and how these results are proven. Our goal is to give an accessible high-level outline of the proofs, but still make clear what the main technical points in the arguments are and how they fit together.

2.1 Substitution Formulas

If F is a CNF formula over variables $x, y, z, \ldots$ and $f : \{0,1\}^d \mapsto \{0,1\}$ is a Boolean function over d variables, we can obtain a new CNF formula by substituting $f(x_1, \ldots, x_d)$ for every variable x (where we assume that $x_1, \ldots, x_d$ are new variables that do not appear anywhere else) and then expand to conjunctive normal form. We will write $F[f]$ to denote the resulting *substitution formula*. For example, for the disjunctive clause $C = x \vee \overline{y}$ and the binary exclusive or function $f(x_1, x_2) = x_1 \oplus_2 x_2$ we have

$$\begin{aligned} C[\oplus_2] = \; & (x_1 \vee x_2 \vee y_1 \vee \overline{y}_2) \wedge (x_1 \vee x_2 \vee \overline{y}_1 \vee y_2) \\ & \wedge (\overline{x}_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2) \enspace . \end{aligned} \tag{2.1}$$

One important observation is that if we hit $F[\oplus_2]$ with a random restriction ρ that sets one of x_1 and x_2 to a random value for every x and leaves the other variable unset, then $F[\oplus_2]\restriction_\rho$ will be the formula F except possibly for sign flips of the literals. It is well known that restrictions also preserve resolution and PCR-refutations, and so for any refutation $\pi : F[\oplus_2] \vdash \bot$ we have that $\pi\restriction_\rho$ is a refutation of F. It is not hard to show that if in addition π has small length/size, then it is likely that $\pi\restriction_\rho$ does not have any wide clauses (in resolution) or high-degree monomials (in PCR). This will be useful in what follows.

2.2 Pebbling Contradictions

Pebbling is a tool for studying time-space relationships by means of a game played on directed acyclic graphs. Pebble games were originally devised for studying programming languages and compiler construction, but have later found a broad range of applications in computational complexity theory, including, during the last decade, proof complexity. An excellent survey of pebbling up to ca 1980 is [Pip80], and some more recent developments are covered in the second author's upcoming survey [Nor12a].

The way pebbling results have been used in proof complexity has mainly been by studying so-called *pebbling contradictions*. These are CNF formulas encoding the pebble game played on a DAG G by postulating the sources to be true and the sink to be false, and specifying that truth propagates through the graph according to the rules of the pebble game.

Definition 2.1 (Pebbling contradiction [BW01]). Suppose that G is a DAG with source vertices S and a unique sink z. Identify every vertex $v \in V(G)$ with a propositional logic variable v. The *pebbling contradiction* over G, denoted Peb_G, is the conjunction of the following clauses:

- for all $s \in S$, a unit clause s (*source axioms*),
- For all non-sources v with predecessors $pred(v)$, the clause $\bigvee_{u \in pred(v)} \overline{u} \vee v$ (*pebbling axioms*),
- for the sink z, the unit clause $\overline{z}$ (*sink axiom*).

For an example of a pebbling contradiction, see the CNF formula in Figure 1(b) defined in terms of the graph in Figure 1(a). If the DAG G has n vertices and maximal indegree ℓ, the formula Peb_G is an unsatisfiable $(1+\ell)$-CNF formula with $n+1$ clauses over n variables.

To make the connection back to Section 2.1, two examples of substituted version of the pebbling formula in Figure 1(b) are the substitution with logical or in Figure 2(a) and with exclusive or in Figure 2(b).

2.3 Substitution Theorem and Trade-offs Based on Pebbling

A paradigm that has turned out to be useful in many contexts in proof complexity is to take a CNF formula family $\{F_n\}_{n=1}^{\infty}$ with interesting properties, tweak it by substituting some function $f(x_1, \ldots, x_d)$ for each variable x as described in Section 2.1, and then use this new formula family to prove the desired result. In particular, the time-space trade-offs in [BN11] fit this pattern. The techniques in [BN11] are developed specifically for resolution and the more general proof system known as k-DNF resolution, but a careful analysis of the proofs reveals that most of the approach can be generalized to other proof systems in a more general setting. We present this general setting below in the hope that it can be useful as an approach for proving space lower bounds and time-space trade-offs for proof systems such as PCR and cutting planes analogous to those for resolution and k-DNF resolution in [BN11]. And indeed, as we shall see soon, a simple special case of this approach combined with random restrictions already yields nontrivial trade-offs for PCR, albeit with some loss in the parameters as compared to the resolution trade-offs in [BN11].

The idea is as follows: Start with a CNF formula F which has a (weak) trade-off between length and variable space in resolution. Consider some proof system $\mathcal{P}$ and study the substitution formula $F[f]$, where we have chosen f to have the right properties with respect to $\mathcal{P}$. Let π_f be any $\mathcal{P}$-refutation of $F[f]$. Intuitively, we want to argue that whatever a $\mathcal{P}$-refutation π_f of $F[f]$ looks like, we can *extract* from this π_f a resolution refutation π of F with related properties. Our way of doing this is to define *projections* of $\mathcal{P}$-configurations over $Vars\big(F[f]\big)$ to clauses over $Vars(F)$, and to show that such projections translate $\mathcal{P}$-refutations to resolution refutations. Roughly, our intuition for projections is that if, for instance, a $\mathcal{P}$-configuration $\mathbb{D}$ implies $f(x_1, \ldots, x_d) \vee \neg f(y_1, \ldots, y_d)$, then this should project the clause $x \vee \overline{y}$. It will be useful for us, however, to relax this requirement a bit and allow other definitions of projections as well as long as they are "in the same spirit." Generalizing [BN11], we show that any function satisfying the following properties will make this approach work.

Definition 2.2 (Projection). Let $f : \{0,1\}^d \mapsto \{0,1\}$ be a fixed Boolean function. Let $\mathcal{P}$ be a sequential proof system, and let $\mathbb{D}$ denote an arbitrary $\mathcal{P}$-configuration over $Vars\big(F[f]\big)$. Let $\mathbb{C}$ denote arbitrary sets of disjunctive clauses over $Vars(F)$. Then the function $proj_f$ mapping $\mathcal{P}$-configurations $\mathbb{D}$ to clauses $\mathbb{C}$ is an *f-projection* if it is:

Complete: If $\mathbb{D} \vDash C[f]$ then the clause C either is in $proj_f(\mathbb{D})$ or is derivable from $proj_f(\mathbb{D})$ by weakening.

Nontrivial: If $\mathbb{D} = \emptyset$, then $proj_f(\mathbb{D}) = \emptyset$.

Monotone: If $\mathbb{D}' \vDash \mathbb{D}$ and $C \in proj_f(\mathbb{D})$, then C is in or is derivable from $proj_f(\mathbb{D}')$ by weakening.

Incrementally sound: Let A be a clause over $Vars(F)$ and let L_A be the encoding of some clause in $A[f]$ as a Boolean function of the type prescribed by $\mathcal{P}$. Then if $C \in proj_f(\mathbb{D} \cup \{L_A\})$, it holds for all literals $a \in Lit(A) \setminus Lit(C)$ that the clause $\overline{a} \vee C$ either is in $proj_f(\mathbb{D})$ or can be derived from $proj_f(\mathbb{D})$ by weakening.

In order for a projection to give interesting results, it should also somehow preserve space when going from the proof system $\mathcal{P}$ to resolution.

Definition 2.3 (Space-faithful projection). We say that $proj_f$ is *space-faithful of degree* K with respect to $\mathcal{P}$ if there is a polynomial Q of degree at most K such that $Q(Sp(\mathbb{D})) \geq \left|Vars(proj_f(\mathbb{D}))\right|$ holds for any $\mathcal{P}$-configuration $\mathbb{D}$ over $Vars\big(F[f]\big)$. We say that $proj_f$ is *exactly space-faithful* if we can choose $Q(x) = x$.

As we will show in Appendix C, if we can define a space-faithful projection for a proof system $\mathcal{P}$ with respect to some space measure in $\mathcal{P}$, then resolution trade-offs between length and variable space in resolution for F are amplified to time-space trade-offs for $F[f]$ in $\mathcal{P}$. Presented in this way, the main technical contribution in [BN11] is proving that certain projections are space-faithful for resolution and k-DNF resolution. Also, this means that in order to prove time-space trade-offs for, say, cutting planes, it would be sufficient to design space-faithful projections for cutting planes as defined above. The trade-offs would then follow by applying the projection machinery in an entirely black-box fashion. Although we do not use the full generality of this machinery in the current paper, we nevertheless believe that the development of this black box is one of our main technical contributions.

Unfortunately, even for PCR and cutting planes it seems challenging to come up with space-faithful projections with respect to the most interesting space measures in these systems. However, there is a specific space measure for which we are able to obtain space-faithful projections for a wide range of proof systems $\mathcal{P}$ (once our refined analysis of [BN11] reveals that this is what we should be aiming for), namely if we consider variable space not only as the "target measure" in resolution but also in $\mathcal{P}$. Furthermore, for this measure we can pick the "substitution function" f to be the identity.

Lemma 2.4. *Let $\mathcal{P}$ be any proof system where any derived line follows semantically from the lines used to derive it*[7] *and fix f to be the identity function. Then there are exactly space-faithful projections from $\mathcal{P}$ to resolution with respect to variable space for any CNF formula F without making substitutions.*

This simple but powerful lemma (which we prove in Appendix C) turns out to be sufficient to lift the resolution trade-offs between width and clause space in [Ben09] to the PCR trade-offs between degree and monomial space in Theorem 1.1 (see Appendix D).

Another simple, but even more powerful, idea is to combine Lemma 2.4 with substitution using exclusive or (over two or more variables). If π is a PCR-refutation of $F[\oplus]$ then after hitting π with a restriction ρ described above, we get a PCR-refutation of the original formula F that is likely not to contain high-degree monomials. But if all monomials are of small degree, then small monomial space implies small variable space, and this means that we can prove a slightly weaker analogue for PCR of the substitution space theorem in [BN11] for resolution, as stated next.

Theorem 2.5 (Substitution space theorem for PCR). *Suppose that F is a CNF formula for which any syntactic resolution refutation in variable space at most s must make more than T axiom downloads.*[8] *Then any semantic PCR-refutation of $F[\oplus]$ in monomial space at most $s/\log_{4/3} T$ must have size larger than T.*

Proof. Suppose that $\pi : F \vdash \bot$ is a PCR-refutation of $F[\oplus]$ in size at most T and in monomial space s'. If we apply a random restriction ρ to $F[\oplus]$ as described above, then $\pi\restriction_\rho$ is a PCR-refutation of F. Consider some fixed monomial m in π. By Lemma B.11, $m\restriction_\rho$ has degree at most K except with probability $(3/4)^K$. Thus,

[7] Note that e.g. extended Frege does *not* satisfy this property, since introducing a new extension variable as a shorthand for a logical formula declares an equivalence that is not the consequence of this formula, but e.g. cutting planes, PC and PCR do.

[8] It would have been nice to be able to use bounds on refutation length here rather than bounds on the number of axiom downloads. This is clearly *not* possible, however. The reason for this is that the proof refuting $F[\oplus]$ is allowed to use any arbitrarily strong *semantic* inference rules, and this can lead to exponential savings compared to syntactic resolution. But happily, the bound in terms of axiom downloads turns out to be exactly what we need for our applications.

by union-bounding we can pick ρ so that $\pi\restriction_\rho$ is a PCR-refutation of F in size at most T, monomial space at most s', and degree at most $\log_{4/3} T$. This means that the variable space of this refutation is upper-bounded by $s' \log_{4/3} T$. Applying the projection in Lemma 2.4, this results in a syntactic resolution proof doing at most T downloads and never exceeding variable space $s' \log_{4/3} T$. This is impossible if $s' \leq s/\log_{4/3} T$, and the theorem follows. □

The time-space trade-offs for PCR in sublinear space reported in Theorems 1.2 and 1.3, as well as several other trade-off results, follow by applying Theorem 2.5 to pebbling formulas substituted with exclusive or. These formulas are all refutable in linear length and constant width simultaneously in resolution, which means that polynomial calculus can simulate these refutations in linear size. In this way, we get trade-off results where the upper bounds hold for syntactic versions of the weaker proof systems resolution and polynomial calculus, whereas the lower bounds hold for the stronger proof system PCR, also when this system is made even stronger by allowing semantic derivation steps.

2.4 Tseitin Contradictions

Let $G = (V, E)$ be a connected undirected graph over $|V| = n$ vertices and let $f : V \mapsto \{0, 1\}$ be a function. If we let each edge $e \in E$ correspond to a variable x_e, we can define local constraints $\bigoplus_{e \ni v} x_e \equiv f(v) \pmod 2$ saying that the parity of the truth values of all edges incident to a vertex v agrees with $f(v)$. We will let $PARITY_v$ denote this constraint encoded in CNF. We say that f has *odd weight* if $\sum f(v) \equiv 1 \pmod 2$. A simple counting argument shows for odd-weight f the collection of parity contraints for all vertices in G is inconsistent.

Definition 2.6 (Tseitin contradiction [Tse68]). For an undirected graph G and an odd-weight function f, the *Tseitin contradiction* over G with respect to f is the CNF formula $Ts(G, f) = \bigwedge_{v \in V(G)} PARITY_v$.

Frequently, we will suppress f and refer to $Ts(G)$ instead, since it can be shown that when G is connected, any two odd-weight functions yield essentially equivalent formulas.

When the degree of the graph is d, each constraint can be written as a CNF formula of 2^d clauses of width d, and hence $Ts(G)$ has $2^d|V|$ clauses in total. Figure 3(b) gives an example Tseitin contradiction generated from the graph in Figure 3(a).

2.5 Time-Space Trade-offs Based on Tseitin Contradictions

Tseitin contradictions are a well-studied family of formulas. A long line of research has shown that isoperimetric properties of the graph G control in some sense the hardness of $Ts(G)$, perhaps explained most clearly in [BW01], which obtains width lower bounds that immediately imply lower bounds for both length and space in resolution (as was discussed briefly in Section 1.1). More recently, [BBI12] used a certain *extended* isoperimetric property crucially to obtain time-space trade-offs, however, for technical reasons this graphs needed to be somewhat dense in the general result, so the resulting CNFs had unbounded width. At a high level this would seem to be unnecessary – the extended isoperimetric property on its own should already imply that short refutations can be roughly divided into a number of levels which are fairly "well-connected" to one another in the sense that the flow of truth assignments is fairly well-distributed. Hard earned intuition coming from the area of Graph Pebbling tells us that DAGs which are layered and have well-connected layers should be difficult to pebble in small space, hence a time space tradeoff result ensues. Of course, the devil is in the details, but the germ of a simpler argument which applies to any graph with extended isoperimetry was already apparent in the Regular Resolution results in [BBI12]. The new argument thus clarifies what was really important in the previous arguments. Also, and regardless of this, a natural question arising from [BBI12] is whether similar results for general resolution could be obtained for graphs

of constant degree, which would yield CNF formulas of constant width (which, as noted in e.g. [ABRW02], is the preferred setting when studying space in proof complexity).

Perhaps somewhat unexpectedly, it turns out that the paradigm of "hardness amplification by substitution" used in [BN11] comes in handy here as well. For our results we study graphs G that are $W \times \ell$ grids, $\ell \gg W$, where every vertex (except those on the boundary of the grid) is connected to its neighbours vertically above and below and horizontally to the left and the right, and then do substitution with binary exlusive or in the Tseitin contradiction to obtain $Ts(G)[\oplus]$. (It is easy to see that for Tseitin formulas, this corresponds to replacing the graph G with a multigraph which has two copies of every edge in G.)

Let's consider a few natural resolution refutations of these formulas. The most obvious approach to refute such a formula is a divide-and-conquer strategy. Suppose we find a small balanced cut in the graph and consider all possible assignments to the variables on that cut. This would divide the problem into 2^W subproblems with half as many vertices. Since the graph is a grid, we can do this repeatedly without finding cuts which are very large. We ultimately obtain a treelike refutation in length at most $2^{O(W \log n)}$ which can be carried out in space $O(W \log n)$. The only thing we are using here is that the grid has the property that we can repeatedly find good cuts.

Frequently, divide-and-conquer solutions to problems have related solutions by dynamic programming. This holds true here as well, and this approach yields a resolution refutation in length $n^{O(1)}2^{O(W)}$ and roughly the same space. These refutations are described in detail in the appendix of [BBI12], but we note that similar results can also be derived using the algoritms of [ACL+12]. These two refutations are the ones appearing in parts 1 and 2 of Theorem 1.4. Part 3 of the theorem then says that the large-space refutation is essentially optimal and that the small-space refutation is at least qualitatively not far from optimal, in that the quasipolynomial blow up in size for small space is unavoidable.

The high level idea in our proof of the lower bound is the same variation on the "bottleneck counting argument" as used in [BBI12]. The bottleneck counting argument was introduced by Haken [Hak85] to give size lower bounds in resolution. The modern form of this argument [BP96] combines a random restriction, which kills off "complex clauses," with a local measure of progress, which proves that any refutation of the formula under study must contain a complex clause at some point in the argument. [BBI12] developed an extension of this argument which assumes a local measure of progress with nice properties and considers how it behaves when a refutation is subdivided into *epochs* recursively and certain small pieces are analyzed; a series of simple averaging arguments show that even after the refutation has been carefully divided into many pieces, in any sound refutation one of these pieces must contain clauses of a large number of different complexities. On the other hand, the extended isoperimetric property makes it straightforward to show that any small collection of clauses in unlikely to contain representatives of many different complexity levels, simply because these clauses will all be fairly wide and so the random restriction has a good chance at eliminating a fair amount of them. If the refutation is too short and uses too little space, a union bound argument show that we can subdivide in such a way that we know for some restriction none of these small pieces will contain enough clauses. This immediately contradicts the fact that after being hit by a restriction, the restricted refutation is proving the unsatisfiability of the corresponding restricted formula.

Morally, it would seem that the same approach should work in PCR, with monomials playing the role of clauses, using familiar restriction arguments. However, constructing an appropriate local measure of progress needed to actually carry out the above argument becomes technically quite challenging. We ultimately side step this difficulty by using a series of simulations inspired by ideas introduced in [BGIP01]. Thus, we are *not* able to define a local measure of progress as described above for PCR. However, we do obtain such a measure in the subsystem of PC in which these simulations are carried out. Using a special property of the simulations, we are able to obtain the monomials of disparate complexities which we need by pushing the restricted proof through this simulation, carrying out the local measure there to find the monomials of disparate complexities within the corresponding epochs and subepochs, and lifting them back through the simulation. The rest of the argument can then be finished as before.

We conclude this overview by describing these simulations in slighly greater detail. The starting point is the observation by Grigoriev [Gri98] that insights concerning the structure of binomial ideals, studied in commutative algebra, can also be useful for proving degree lower bounds for the Nullstellensatz proof system, a restricted form of PC. In [BGIP01], these ideas were extended to show degree lower bounds for polynomial calculus. A long line of work [CEI96, IPS99, Raz98, Raz01, AR03, Gri98, BGIP01] on degree lower bounds has proceeded by constructing an explicit "pseudoideal," that is, a set of polynomials closed under the derivation rules of PC as long as the degree d is bounded, and which contains all the axioms of the formula. Such a set clearly contains all polynomials provable in degree d, and the lower bound is established by showing that contradiction, i.e., the constant term 1, is not in this set. [BGIP01] gave a simple method to obtain explicit pseudoideals when all the axioms of the formula as encoded in PC are *binomials*. The heart of the technique is a clever algebraic manipulation to show that the information which can be obtained in low degree can also be obtained by using only proof lines that are binomials. They also gave "low degree reductions" which showed that Tseitin contradictions, among other formulas, can be translated without loss of generality, as far as degree is concerned, to systems of binomials, by means of what is essentially the Fourier transform. We reinterpret both of these steps as simulations explicitly and observe that their efficiency with respect to degree is a byproduct of being "conservative with respect to monomials". This simple property is exactly what we need to lift complex monomials from the simulated binomial refutations to our original refutations and overcome the chief technical obstacle alluded to above. We find it somewhat surprising that this turns out to be useful for time-space trade-offs, because in general the combination of these two simulations are wildly inefficient with respect to proof size and so we lose control over that measure as soon as the simulations are applied. We fill in the details in the proof outline above in Appendix G.

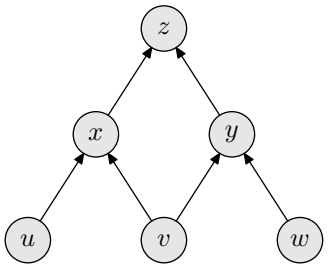
A Concluding Remarks

In this paper, we report the first trade-off results for polynomial calculus (and PCR) which rule out simultaneous optimization of different proof complexity measures, in particular proof size and proof space. Loosely speaking, what our results say is that in the worst case, it is impossible to do any meaningful simultaneous optimization of size and space in polynomial calculus.

Polynomial calculus and PCR are still not very well understood proof systems, however, and there remain a number of open problems. For instance, it is known that proof size in PC is at most exponential, and there are also exponential lower bounds for PCR. Space in PC is also at most linear in the worst case if the formulas have bounded width [FLN+12], but here no matching lower bounds are known. It seems natural to expect that space in PC and PCR should indeed be linear in the worst case (for instance, for random k-CNF formulas), and it would be very interesting if this could be proven (or ruled out, but this would arguably be quite surprising).

In resolution, we know that clause space is an upper bound on width [AD08], that small width does not say anything about space complexity [BN08], and that there can be very strong trade-offs between these two measures [Ben09]. In this paper, we have shown that exactly the same kind of trade-off holds between degree and monomial space in PC and PCR, but we do not know whether monomial space is an upper bound on degree or whether small degree says anything about the space complexity.

For our time-space trade-off results in sublinear space using pebbling formulas, it would be very satisfying to remove the loss in the parameters resulting from having to take the logarithm of the proof size. This loss is inherent in the restriction argument, but for resolution it is known how to avoid restrictions completely and instead use the projection machinery in Appendix C together with the right kind of substitutions in the formulas to get tight trade-offs. It would be very satisfying if something similar could be made to work for PC and PCR, since this would give tight trade-offs (for sublinear space) for these two proof systems and



(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 1: Example pebbling contradiction.

also potentially yield stronger unconditional space lower bounds than are currently known.

Looking beyond polynomial calculus, another proof system that would be very interesting to understand is cutting planes. Here the open questions abound. Perhaps most obviously, it would be desirable to prove size lower bounds by some other technique than the interpolation used in [Pud97], for instance, for Tsetin formulas or random k-CNF formulas.

As far as we are aware, there are no space lower bounds or "true" time-space trade-offs known for cutting planes. However, the recent results in [HN12] could be interpreted to suggest that pebbling formulas of the right flavour should inherit time-space trade-offs properties from the graphs in terms of which they are defined not only for the resolution proof system but also extending to cutting planes. If true, this would mean that the so-called black-white pebble game in [CS76] could be used to obtain strong trade-offs not only for resolution, k-DNF resolution and PC/PCR, but also for cutting planes.

Finally, let us note that it is known that PCR and cutting planes are both strictly stronger than resolution with respect to proof size. It seems natural to expect that PCR and cutting planes should both be stronger than resolution with respect to space as well, but as far as we are aware this is open. Thus, it would be nice to separate PCR from resolution with respect to space by finding a k-CNF formula that has low monomial space complexity in PCR but large clause space complexity in resolution, and similarly for cutting planes with respect to resolution.

Acknowledgements

This article is the result of a long process, and various subsets of the authors would like to acknowledge useful discussions had during the last few years with various subsets of Paul Beame, Eli Ben-Sasson, Arkadev Chattopadhyay, Yuval Filmus, Trinh Huynh, Russell Impagliazzo, Massimo Lauria, Alexander Razborov, and Noga Zewi.

The work presented in this paper was initiated at the Banff International Research Station workshop on proof complexity (11w5103) in October 2011. Part of the work was also carried out at KTH Royal Institute of Technology where the visits of the first and third authors were partially supported by the foundations *Johan och Jakob Söderbergs stiftelse*, *Magnus Bergvalls Stiftelse*, *Stiftelsen Längmanska kulturfonden*, and *Helge Ax:son Johnsons stiftelse*.

The research of the first author was supported by NSF grant CCF-0832797. The second author was supported by Swedish Research Council grant 621-2010-4797 and by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no 279611. The third author was supported in part by the National Basic Research Program of China Grant

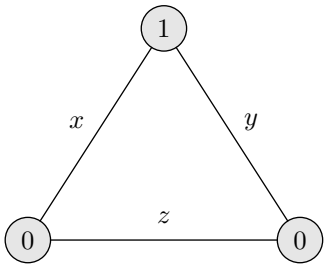
$$
\begin{array}{ll}
(u_1 \vee u_2) & \wedge (\overline{v}_2 \vee \overline{w}_1 \vee y_1 \vee y_2) \\
\wedge (v_1 \vee v_2) & \wedge (\overline{v}_2 \vee \overline{w}_2 \vee y_1 \vee y_2) \\
\wedge (w_1 \vee w_2) & \wedge (\overline{x}_1 \vee \overline{y}_1 \vee z_1 \vee z_2) \\
\wedge (\overline{u}_1 \vee \overline{v}_1 \vee x_1 \vee x_2) & \wedge (\overline{x}_1 \vee \overline{y}_2 \vee z_1 \vee z_2) \\
\wedge (\overline{u}_1 \vee \overline{v}_2 \vee x_1 \vee x_2) & \wedge (\overline{x}_2 \vee \overline{y}_1 \vee z_1 \vee z_2) \\
\wedge (\overline{u}_2 \vee \overline{v}_1 \vee x_1 \vee x_2) & \wedge (\overline{x}_2 \vee \overline{y}_2 \vee z_1 \vee z_2) \\
\wedge (\overline{u}_2 \vee \overline{v}_2 \vee x_1 \vee x_2) & \wedge \overline{z}_1 \\
\wedge (\overline{v}_1 \vee \overline{w}_1 \vee y_1 \vee y_2) & \wedge \overline{z}_2 \\
\wedge (\overline{v}_1 \vee \overline{w}_2 \vee y_1 \vee y_2) &
\end{array}
$$

(a) Substitution pebbling contradiction $Peb_{\Pi_2}[\vee_2]$ with respect to binary logical or.

$$
\begin{array}{ll}
(u_1 \vee u_2) & \wedge (v_1 \vee \overline{v}_2 \vee \overline{w}_1 \vee w_2 \vee y_1 \vee y_2) \\
\wedge (\overline{u}_1 \vee \overline{u}_2) & \wedge (v_1 \vee \overline{v}_2 \vee \overline{w}_1 \vee w_2 \vee \overline{y}_1 \vee \overline{y}_2) \\
\wedge (v_1 \vee v_2) & \wedge (\overline{v}_1 \vee v_2 \vee w_1 \vee \overline{w}_2 \vee y_1 \vee y_2) \\
\wedge (\overline{v}_1 \vee \overline{v}_2) & \wedge (\overline{v}_1 \vee v_2 \vee w_1 \vee \overline{w}_2 \vee \overline{y}_1 \vee \overline{y}_2) \\
\wedge (w_1 \vee w_2) & \wedge (\overline{v}_1 \vee v_2 \vee \overline{w}_1 \vee w_2 \vee y_1 \vee y_2) \\
\wedge (\overline{w}_1 \vee \overline{w}_2) & \wedge (\overline{v}_1 \vee v_2 \vee \overline{w}_1 \vee w_2 \vee \overline{y}_1 \vee \overline{y}_2) \\
\wedge (u_1 \vee \overline{u}_2 \vee v_1 \vee \overline{v}_2 \vee x_1 \vee x_2) & \wedge (x_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2 \vee z_1 \vee z_2) \\
\wedge (u_1 \vee \overline{u}_2 \vee v_1 \vee \overline{v}_2 \vee \overline{x}_1 \vee \overline{x}_2) & \wedge (x_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2 \vee \overline{z}_1 \vee \overline{z}_2) \\
\wedge (u_1 \vee \overline{u}_2 \vee \overline{v}_1 \vee v_2 \vee x_1 \vee x_2) & \wedge (x_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2 \vee z_1 \vee z_2) \\
\wedge (u_1 \vee \overline{u}_2 \vee \overline{v}_1 \vee v_2 \vee \overline{x}_1 \vee \overline{x}_2) & \wedge (x_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2 \vee \overline{z}_1 \vee \overline{z}_2) \\
\wedge (\overline{u}_1 \vee u_2 \vee v_1 \vee \overline{v}_2 \vee x_1 \vee x_2) & \wedge (\overline{x}_1 \vee x_2 \vee y_1 \vee \overline{y}_2 \vee z_1 \vee z_2) \\
\wedge (\overline{u}_1 \vee u_2 \vee v_1 \vee \overline{v}_2 \vee \overline{x}_1 \vee \overline{x}_2) & \wedge (\overline{x}_1 \vee x_2 \vee y_1 \vee \overline{y}_2 \vee \overline{z}_1 \vee \overline{z}_2) \\
\wedge (\overline{u}_1 \vee u_2 \vee \overline{v}_1 \vee v_2 \vee x_1 \vee x_2) & \wedge (\overline{x}_1 \vee x_2 \vee \overline{y}_1 \vee y_2 \vee z_1 \vee z_2) \\
\wedge (\overline{u}_1 \vee u_2 \vee \overline{v}_1 \vee v_2 \vee \overline{x}_1 \vee \overline{x}_2) & \wedge (\overline{x}_1 \vee x_2 \vee \overline{y}_1 \vee y_2 \vee \overline{z}_1 \vee \overline{z}_2) \\
\wedge (v_1 \vee \overline{v}_2 \vee w_1 \vee \overline{w}_2 \vee y_1 \vee y_2) & \wedge (z_1 \vee \overline{z}_2) \\
\wedge (v_1 \vee \overline{v}_2 \vee w_1 \vee \overline{w}_2 \vee \overline{y}_1 \vee \overline{y}_2) & \wedge (\overline{z}_1 \vee z_2)
\end{array}
$$

(b) Substitution pebbling contradiction $Peb_{\Pi_2}[\oplus_2]$ with respect to binary exclusive or.

Figure 2: Examples of substitution pebbling formulas for the pyramid graph Π_2.



(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 contradiction.

Figure 3: Example Tseitin contradiction.

2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61061130540, 61073174.

B Preliminaries

For x a Boolean variable, a *literal over* x is either the variable x itself, called a *positive literal over* x, or its negation, denoted $\neg x$ or $\overline{x}$ and called a *negative literal over* 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. Below we will think of clauses and terms as sets, so that the ordering of the literals is inconsequential and that, in particular, no literals are repeated. A clause (term) containing at most k literals is called a *k-clause* (*k-term*). A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses, and a *DNF formula* is a disjunction of terms. We will think of CNF and DNF formulas as sets of clauses and terms, respectively. A *k-CNF formula* is a CNF formula consisting of k-clauses, and a *k-DNF formula* consists of k-terms.

The *variable set* of a clause C, denoted $\mathit{Vars}(C)$, is the set of Boolean variables over which there are literals in C, and we write $\mathit{Lit}(C)$ to denote the set of literals in C. The variable and literal sets of a term are similarly defined and these definitions are extended to CNF and DNF formulas by taking unions. If V is a set of Boolean variables and $\mathit{Vars}(C) \subseteq V$ we say C is a clause *over* V and similarly define terms, CNF formulas, and DNF formulas over V.

We write α, β to denote truth value assignments. Truth value assignments are functions to $\{0, 1\}$, where we identify 0 with false and 1 with true. We have the usual semantics that a clause is true under α, or *satisfied* by α, if at least one literal in it is true, and a term is true if all literals evaluate to true. We write $\bot$ to denote the empty clause without literals that is false under all truth value assignments. A CNF formula is satisfied if all clauses in it are satisfied, and for a DNF formula we require that some term should be satisfied. In general, we will not distinguish between a formula and the Boolean function computed by it.

If $\mathbb{C}$ is a set of Boolean functions we say that an assignment satisfies $\mathbb{C}$ if and only if it satisfies every function in $\mathbb{C}$. For $\mathbb{D}, \mathbb{C}$ two sets of Boolean functions over a set of variables V, we say that $\mathbb{D}$ *implies* $\mathbb{C}$, denoted $\mathbb{D} \models \mathbb{C}$, if and only if every assignment $\alpha : V \mapsto \{0, 1\}$ that satisfies $\mathbb{D}$ also satisfies $\mathbb{C}$. In particular, $\mathbb{D} \models \bot$ if and only if $\mathbb{D}$ is *unsatisfiable* or *contradictory*, i.e., if no assignment satisfies $\mathbb{D}$. If a CNF formula F is unsatisfiable but for any clause $C \in F$ it holds that the clause set $F \setminus \{C\}$ is satisfiable, we say that F is *minimally unsatisfiable*.

We say that a proof system is *sequential* if a proof π in the system is a *sequence* of lines $\pi = \{L_1, \ldots, L_\tau\}$ of some prescribed syntactic form depending on the proof system in question, where each line is derived from previous lines by one of a finite set of allowed *inference rules*. Following the exposition in [ET01], we view a proof as similar to a non-deterministic Turing machine computation, with a special read-only input

tape from which the clauses of the CNF formula F being refuted (the *axioms*) can be downloaded and a working memory where all derivation steps are made. Then the length of a proof is essentially the time of the computation and space measures memory consumption. The following definition is a straightforward generalization of [ABRW02].

Definition B.1 (Refutation). For a sequential proof system $\mathcal{P}$, a $\mathcal{P}$-con*figuration* $\mathbb{D}$ is a set of lines L of the syntactic form prescribed by $\mathcal{P}$. A sequence of configurations $\{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ is a *$\mathcal{P}$-derivation* from a CNF formula F if $\mathbb{D} = \emptyset$ and for all $t \in [\tau]$, the set $\mathbb{D}_t$ is obtained from $\mathbb{D}_{t-1}$ by one of the following *derivation steps*:

Axiom Download $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L_C\}$, where L_C is the encoding of a clause $C \in F$ in the syntactic form prescribed by the proof system (an *axiom*).

Inference $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L\}$ for some L inferred by one of the inference rules for $\mathcal{P}$ from a set of assumptions $L_1, \ldots, L_m \in \mathbb{D}_{t-1}$.

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

A $\mathcal{P}$-refutation $\pi : F \vdash \bot$ of a CNF formula F is a derivation $\pi = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ such that $\mathbb{D}_0 = \emptyset$ and $\bot \in \mathbb{D}_\tau$, where $\bot$ is the representation of contradiction (e.g. for resolution and $\mathcal{R}(k)$-systems the empty clause without literals).

If every line L in a derivation is used at most once before being erased (though it can possibly be rederived later), we say that the derivation is *tree-like*. This corresponds to changing the inference rule so that $L_1, \ldots, L_d$ must all be erased after they have been used to derive L.

To every refutation π we can associate a DAG G_π, with the lines in π labelling the vertices and with edges from the assumptions to the consequence for each application of an inference rule. There might be several different derivations of a line L during the course of the refutation π, but if so we can label each occurrence of L with a time-stamp when it was derived and keep track of which copy of L is used where. Using this representation, a refutation π can be seen to be tree-like if G_π is a tree.

Definition B.2 (Refutation size, length and space). Given a size measure $S(L)$ for lines L in $\mathcal{P}$-derivations (which we usually think of as the number of symbols in L, but other definitions can also be appropriate depending on the context), the *size* of a $\mathcal{P}$-derivation π is the sum of the sizes of all lines in a derivation, where lines that appear multiple times are counted with repetitions (once for every vertex in G_π). The *length* of a $\mathcal{P}$-derivation π is the number of axiom downloads and inference steps in it, i.e., the number of vertices in G_π.[9] For a space measure $Sp_\mathcal{P}(\mathbb{D})$ defined for $\mathcal{P}$-configurations, the *space* of a derivation π is defined as the maximal space of a configuration in π.

If π is a refutation of a formula F in size S and space s, then we say that F can be refuted in size S and space s *simultaneously*. Similarly, F can be refuted in length L and space s simultaneously if there is a $\mathcal{P}$-refutation $\mathcal{P}$ with $L(\pi) = L$ and $Sp(\pi) = s$.

We define the *$\mathcal{P}$-refutation size* of a formula F, denoted $S_\mathcal{P}(F \vdash \bot)$, to be the minimum size of any $\mathcal{P}$-refutation of it. The *$\mathcal{P}$-refutation length* $L_\mathcal{P}(F \vdash \bot)$ and *$\mathcal{P}$-refutation space* $Sp_\mathcal{P}(F \vdash \bot)$ of F are analogously defined by taking the minimum with respect to length or space, respectively, over all $\mathcal{P}$-refutations of F.

When the proof system in question is clear from context, we will drop the subindex in the proof complexity measures.

[9] The reader who so prefers can instead define the length of a derivation $\pi = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ as the number of steps τ in it, since the difference is at most a factor of 2. We have chosen the definition above for consistency with previous papers defining length as the number of lines in a listing of the derivation.

Let us next give formal definitions in the framework of Definition B.1 of the proof systems that will be of interest in this paper. Below, the notation $\frac{G_1 \quad \cdots \quad G_m}{H}$ means that if $G_1, \ldots, G_m$ have been derived previously in the proof (and are currently in memory), then we can infer H. We will sometimes use notation $G_1, \ldots, G_m \vdash H$ for this as well, for convenience.

Definition B.3 (k-DNF resolution). The *k-DNF resolution* proof systems are a family of sequential proof systems $\mathcal{R}(k)$ parameterized by $k \in \mathbb{N}^+$. Lines in a k-DNF-resolution refutation are k-DNF formulas and we have the following inference rules (where G, H denote k-DNF formulas, T, T' denote k-terms, and $a_1, \ldots, a_k$ denote literals):

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

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

∧-elimination $\dfrac{G \vee T}{G \vee T'}$ for any $T' \subseteq T$.

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

For standard resolution, i.e., $\mathcal{R}(1)$, the k-cut rule simplifies to the *resolution rule*

$$\frac{B \vee x \qquad C \vee \overline{x}}{B \vee C} \tag{B.1}$$

for clauses B and C. We refer to (B.1) as *resolution on the variable* x and to $B \vee C$ as the *resolvent* of $B \vee x$ and $C \vee \overline{x}$ on x. Clearly, in resolution the $\wedge$-introduction and $\wedge$-elimination rules do not apply. It can also be shown that the weakening rule never needs to be used in resolution refutations, but it can be convenient to allow it to simplify some technical arguments in proofs.

For $\mathcal{R}(k)$-systems, the length measure is as defined in Definition B.2, and for space we get the two measures *formula space* and *total space* depending on whether we consider the number of k-DNF formulas in a configuration or all literals in it, counted with repetitions. For standard resolution there are two more space-related measures that will be relevant, namely *width* and *variable space*. For clarity, let us give an explicit definition of all space-related measures for resolution that will be of interest.

Definition B.4 (Width and space in resolution). The *width* $W(C)$ of a clause C is the number of literals in it, and the width of a CNF formula or clause configuration is the size of a widest clause in it. The *clause space* (as the formula space measure is known in resolution) $Sp(\mathbb{C})$ of a clause configuration $\mathbb{C}$ is $|\mathbb{C}|$, i.e., the number of clauses in $\mathbb{C}$, the *variable space*[10] $VarSp(\mathbb{C})$ is $|Vars(C)|$, i.e., the number of distinct variables mentioned in $\mathbb{C}$, and the *total space* $TotSp(\mathbb{C})$ is $\sum_{C \in \mathbb{C}} |C|$, i.e., the total number of literals in $\mathbb{C}$ counted with repetitions.

The width or space of a resolution refutation π is the maximum that the corresponding measures attain over any clause configuration $\mathbb{C} \in \pi$, and taking the minimum over all resolution refutations of a CNF formula F, we can define the width $W_{\mathcal{R}}(F \vdash \bot) = \min_{\pi : F \vdash \bot}\{W(\pi)\}$ of refuting F in resolution, and analogously the clause space $Sp_{\mathcal{R}}(F \vdash \bot) = \min_{\pi : F \vdash \bot}\{Sp(\pi)\}$, variable space $VarSp_{\mathcal{R}}(F \vdash \bot) = \min_{\pi : F \vdash \bot}\{VarSp(\pi)\}$, and total space $TotSp_{\mathcal{R}}(F \vdash \bot) = \min_{\pi : F \vdash \bot}\{TotSp(\pi)\}$ of refuting F.

[10] It should be noted that there is some terminological confusion in the literature here. The term "variable space" has also been used previously to refer to what is here called "total space." The terminology adopted in this paper is due to Alex Hertel and Alasdair Urquhart (see [Her08]), and we feel that their naming convention is the most natural one.

Remark B.5. When studying and comparing the complexity measures for resolution in Definition B.4, as was noted in [ABRW02] it is preferable to prove the results for k-CNF formulas, i.e., formulas where all clauses have size upper-bounded by some constant. This is so since the width and space measures can "misbehave" rather artificially for formula families of unbounded width (see [Nor09b, Section 5] for a discussion of this). Since every CNF formula can be rewritten as an equivalent formula of bounded width by using auxiliary variables, it therefore seems natural to insist that the formulas under study should have width bounded by some constant.

Polynomial calculus (PC), was introduced in [CEI96], though that paper used the name "Gröbner proof system." In a PC-refutation, clauses are interpreted as multilinear polynomials. For instance, the requirement that the clause $x \vee y \vee \overline{z}$ should be satisfied gets translated to the equation $(1-x)(1-y)z = 0$ or $xyz - xz - yz + z = 0$, and we derive contradiction by showing that there is no common root for the polynomial equations corresponding to all the clauses.[11]

Definition B.6 (Polynomial calculus (PC)). In a polynomial calculus proof, lines are multivariate polynomial equations $p = 0$, where $p \in \mathbb{F}[x, y, z, \ldots]$ for some (fixed) field $\mathbb{F}$. It is customary to omit "$= 0$" and only write p. The derivation rules are as follows, where $\alpha, \beta \in \mathbb{F}$, $p, q \in \mathbb{F}[x, y, z, \ldots]$, and x is any variable:

Linear combination $\dfrac{p \qquad q}{\alpha p + \beta q}$

Multiplication $\dfrac{p}{xp}$

A PC-refutation ends when 1 has been derived (i.e., $1 = 0$).

In the context of SAT Solving and also in most Proof Complexity Scenarios, PC also makes use of the following axioms:

Boolean axioms $\dfrac{}{x^2 - x}$ (forcing 0/1-solutions).

The *size* of a PC-refutation is defined as the total number of monomials in the refutation, the *length* of a refutation is the number of polynomial equations, and the *(monomial) space* is the maximal number of monomials in any configuration (counted with repetitions). Another important measure is the *degree* of a refutation, which is the maximal (total) degree of any monomial.

The representation of a clause $\bigvee_{i=1}^{n} x_i$ as a PC-polynomial is $\prod_{i=1}^{n}(1 - x_i)$, which means that the number of monomials is exponential in the clause width. This problem arises only for positive literals, however—a large clause with only negative literals is translated to a single monomial. This is a weakness of monomial space in polynomial calculus when compared to clause space in resolution. In order to obtain a cleaner, more symmetric treatment of proof space, in [ABRW02] the proof system *polynomial calculus resolution (PCR)* was introduced as a common extension of polynomial calculus and resolution. The idea is to add an extra set of parallell formal variables $x', y', z', \ldots$ so that positive and negative literals can both be represented in a space-efficient fashion.

Definition B.7 (Polynomial calculus resolution (PCR)). Lines in a PCR-proof are polynomials over the ring $\mathbb{F}[x, x', y, y', z, z', \ldots]$, where as before $\mathbb{F}$ is some field. We have all the axioms and rules of PC plus the following axioms:

Complementarity $\dfrac{}{x + x' - 1}$ for all pairs of variables (x, x').

[11] In fact, from a mathematical point of view it seems more natural to think of 0 as true and 1 as false in polynomial calculus, so that the unit clause x gets translated to $x = 0$. For simplicity and consistency in this paper, however, we stick to thinking about $x = 1$ as meaning that x is true and $x = 0$ as meaning that x is false.

Size, length, and degree are defined as for polynomial calculus, and the (monomial) space of a PCR-refutation is again the maximal number of monomials in any configuration counted with repetitions.[12]

The point of the complementarity rule is to force x and x' to have opposite values in $\{0,1\}$, so that they encode complementary literals. This means one can potentially avoid an exponential blow-up in size measured in the number of monomials (and thus also for space). Our running example clause $x \vee y \vee \overline{z}$ is rendered as $x'y'z$ in PCR. In PCR, monomial space is a natural generalization of clause space since every clause translates into a monomial as just explained.

It's convenient to define the following subsystem of PC.

Definition B.8 (Binomial PC). A *binomial* is a sum of two monomials, one or both of which may be zero. A system of equations will be called a *binomial system* if all of the polynomial constraints $p = 0$ are binomials. A *Binomial PC* derivation is one in which for every proof line $p = 0$, p is a binomial.

Because of the quite restrictive constraint that all axioms be Binomials, Binomial PC is studied without the Boolean Axioms; as discussed in [BGIP01], Binomial PC with Boolean Axioms can be simulated efficiently by Resolution. Thus, Binomial PC shouldn't be thought of as directly useful to SAT solving model, but more typically as a proof system to reduce to; if a good change of variables permits one to rephrase a formula as a binomial system it may be profitable to study it via Binomial PC. It's also worth pointing out that there really can be no binomial PCR, since the extension axioms $\overline{x_i} = 1 + x_i$ are not binomials.

Definition B.9. A *restriction* here is a mapping ρ from some set of boolean variables to constants and other variables. The restriction of a formula F, denoted by $F\restriction_\rho$, is the formula obtained by replacing the variables with their images under the restriction and performing local simplifications. The restriction of a set of formulas is defined by restricting each one. The restriction of a polynomial p, denoted by $p\restriction_\rho$ is defined similarly. Notice that allowing a variable being mapped to another variable is a bit non-standard. Nevertheless, this non-standardness does not affect a fact that is well known for restrictions allowing mapping only to constants (a.k.a. partial assignments): proofs may also be restricted – if π is a proof of F, and ρ is a restriction, then there is an induced proof $\pi\restriction_\rho$ of $F\restriction_\rho$, obtained again by performing local simplifications to the formulas of the proof and to the proof structure.

For any formula F, there is a generic distribution of random restrictions to the variables $F[\oplus]$ which is often useful.

Definition B.10. Let F be any formula. Define a random restriction ρ which maps variables of $F[\oplus]$ to constants and variables of F by independently for each variable x of F, choosing one of x_1 and x_2 with equal probability to set to $\{0,1\}$ with equal probability, and the other to either x or $\overline{x}$ so that $\rho(x_1) \oplus \rho(x_2) \equiv x$.

This restriction sets a variable to a constant with constant probability, so it is a fairly dense restriction which can be used to kill very wide clauses. On the other hand it is very clean and always gives us back essentially the same formula. Note that it always holds that $F[\oplus]\restriction_\rho = F$.

We'll use this next lemma throughout the paper, and throughout the paper ρ will refer to a random restriction chosen in this way.

Lemma B.11. *Let C be any clause in the variables of $F[\oplus]$. Then,*

$$\Pr_\rho\left[\left|\mathit{Vars}(C\restriction_\rho)\right| \geq K\right] \leq \left(\frac{3}{4}\right)^K .$$

[12] We remark that in [ABRW02] space was defined as the number of *distinct* monomials in a configuration (i.e., not counted with repetitions), but we find this restriction to be somewhat arbitrary.

Proof. Suppose $\left|\{x : x_1, x_2 \in Vars(C\restriction_\rho)\}\right| < K$. Then the probability of the event is zero. Suppose not. Then there are at least K vars of C which ρ assigns independently, to 0 with probability at least $1/4$ and to 1 with probability at least $1/4$. Since for each such variable, there is a specific value which if chosen by ρ will result in $C\restriction_\rho = \top$ and $Vars(C\restriction_\rho) = \emptyset$. Therefore by independence, the probability that $Vars(C\restriction_\rho) \neq \emptyset$ is at most $\left(\frac{3}{4}\right)^K$, as desired. □

Note that while in some specific applications better results can be achieved with specially crafted random restrictions, if we don't care about constants in the exponent this kind of argument can often yield optimal results with a very simple proof. This technique has the advantage that often we won't need to think about $F[\oplus]$ directly and can instead focus on F.

In general, the admissible inferences in a proof system according to Definition B.1 are defined by a set of syntactic inference rules. In what follows, we will also be interested in a strengthened version of this concept, which was made explicit in [ABRW02].

Definition B.12 (Syntactic and semantic derivations). We refer to derivations according to Definition B.1, where each new line L has to be inferred by one of the inference rules for $\mathcal{P}$, as *syntactic* derivations. If instead *any line* L that is semantically implied by the current configuration can be derived in one atomic step, we talk about a *semantic* derivation.

Clearly, semantic derivations are at least as strong as syntactic ones, and they are easily seen to be superpolynomially stronger with respect to length for any proof system where superpolynomial lower bounds are known. This is so since a semantic proof system can download all axioms in the formula one by one, and then deduce contradiction in one step since the formula is unsatisfiable. Therefore, semantic versions of proof systems are mainly interesting when we want to reason about space or the relationship between space and length. But if we can prove lower bounds not just for syntactic but even semantic versions of proof systems, this of course makes these bounds much stronger.

Let us finally remark that although the measure of total space, considering the total number of symbols in memory, is perhaps a priori the most natural one, most papers on proof space have focused on space measured as the number of lines in memory (e.g., clauses, k-DNF formulas, or inequalities). However, as observed in [ABRW02], for strong enough proof systems, this "line space" measure is no longer interesting since just one unit of memory can contain a big AND of all formulas derived so far. The "line space" measure makes perfect sense for resolution and k-DNF resolution, and seems to do so also for cutting planes. For PC/PCR, however, measuring just the number of polynomial equations is not very meaningful, since every equation can be of exponential size and encode very much information. Instead, the natural generalization of clause space is monomial space.

C Substituted Formulas, Projections, and Trade-offs

Let us start by recalling some key definitions from Section 2.3.

Definition 2.2 (restated). Let $f : \{0,1\}^d \mapsto \{0,1\}$ be a fixed Boolean function. Let $\mathcal{P}$ be a sequential proof system, and let $\mathbb{D}$ denote an arbitrary $\mathcal{P}$-configuration over $Vars\big(F[f]\big)$. Let $\mathbb{C}$ denote arbitrary sets of disjunctive clauses over $Vars(F)$. Then the function $proj_f$ mapping $\mathcal{P}$-configurations $\mathbb{D}$ to clauses $\mathbb{C}$ is an *f-projection* if it is:

Complete: If $\mathbb{D} \vDash C[f]$ then the clause C either is in $proj_f(\mathbb{D})$ or is derivable from $proj_f(\mathbb{D})$ by weakening.

Nontrivial: If $\mathbb{D} = \emptyset$, then $proj_f(\mathbb{D}) = \emptyset$.

Monotone: If $\mathbb{D}' \vDash \mathbb{D}$ and $C \in \mathit{proj}_f(\mathbb{D})$, then either $C \in \mathit{proj}_f(\mathbb{D}')$ or C is derivable from $\mathit{proj}_f(\mathbb{D}')$ by weakening.

Incrementally sound: Let A be a clause over $\mathit{Vars}(F)$ and let L_A be the encoding of some clause in $A[f]$ as a Boolean function of the type prescribed by $\mathcal{P}$. Then if $C \in \mathit{proj}_f(\mathbb{D} \cup \{L_A\})$, it holds for all literals $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ that the clause $\overline{a} \vee C$ either is in $\mathit{proj}_f(\mathbb{D})$ or can be derived from $\mathit{proj}_f(\mathbb{D})$ by weakening.

Definition 2.3 (restated). Consider a sequential proof system $\mathcal{P}$ with space measure $\mathit{Sp}(\cdot)$. Suppose that $f : \{0,1\}^d \mapsto \{0,1\}$ is a fixed Boolean function, and that proj_f is an f-projection. Then we say that proj_f is *space-faithful of degree* K with respect to $\mathcal{P}$ if there is a polynomial Q of degree at most K such that $Q(\mathit{Sp}(\mathbb{D})) \geq \left|\mathit{Vars}(\mathit{proj}_f(\mathbb{D}))\right|$ holds for any $\mathcal{P}$-configuration $\mathbb{D}$ over $\mathit{Vars}\big(F[f]\big)$. We say that proj_f is *linearly space-faithful* if Q has degree 1, and that proj_f is *exactly space-faithful* if we can choose $Q(x) = x$.

A special kind of projections are those that look not only on all of $\mathbb{D}$ "globally," but measure the semantic content of $\mathbb{D}$ more precisely.

Definition C.1 (Local projection). If proj_f is an f-projection, then its *localized version* proj_f^L is defined to be $\mathit{proj}_f^L(\mathbb{D}) = \bigcup_{\mathbb{D}' \subseteq \mathbb{D}} \mathit{proj}_f(\mathbb{D}')$. If $\mathit{proj}_f = \mathit{proj}_f^L$, we say that proj_f is a *local projection*.

It is easily verified that the localized version of a projection is indeed itself a projection in the sense of Definition 2.2.

C.1 Using Projections to Obtain Time-Space Trade-offs

We now show that if we can design a projection in accordance with Definition 2.2, then this projection can be used to extract resolution refutations from $\mathcal{P}$-refutations. Furthermore, if our projection is space-faithful, this extraction operation will preserve length-space trade-off (with some loss in parameters depending on how high the degree K is). First we need a definition restricting what kind of sequential proof systems we are considering.

Definition C.2 (Implicational proof system). Let us say that $\mathcal{P}$ is an *implicational* sequential proof system if any derived line follows semantically from the lines used to derive it.

We remark that resolution, PC, PCR, cutting planes, Frege and most other proof systems usually studied are "implicational" in the sense of Definition C.2, where as, for instance, extended Frege is not.

Lemma C.3. *Let $\mathcal{P}$ be an implicational sequential proof system and let $f : \{0,1\}^d \mapsto \{0,1\}$ be a Boolean function, and suppose that proj_f is an f-projection. Then for any CNF formula F it holds that if $\pi_f = \{\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau\}$ is a semantic $\mathcal{P}$-refutation of the substitution formula $F[f]$, the sequence of sets of projected clauses $\big\{\mathit{proj}_f(\mathbb{D}_0), \mathit{proj}_f(\mathbb{D}_1), \ldots, \mathit{proj}_f(\mathbb{D}_\tau)\big\}$ forms the "backbone" of a resolution refutation π of F in the following sense:*

1. $\mathit{proj}_f(\mathbb{D}_0) = \emptyset$.
2. $\bot \in \mathit{proj}_f(\mathbb{D}_\tau)$.
3. *All transitions from $\mathit{proj}_f(\mathbb{D}_{t-1})$ to $\mathit{proj}_f(\mathbb{D}_t)$ for $t \in [\tau]$ can be accomplished in syntactic resolution in such a fashion that $\mathit{VarSp}(\pi) = \mathrm{O}\big(\max_{\mathbb{D} \in \pi_f}\{\mathit{VarSp}(\mathit{proj}_f(\mathbb{D}))\}\big)$, or, if proj_f is a local projection, so that $\mathit{VarSp}(\pi) \leq \max_{\mathbb{D} \in \pi_f}\{\mathit{VarSp}(\mathit{proj}_f(\mathbb{D}))\}$.*
4. *The length of π is upper-bounded by π_f in the sense that the only time π performs a download of an axiom $C \in F$ is when π_f downloads some axiom $D \in C[f]$ from $F[f]$.*

On the one hand, Lemma C.3 is very strong in the sense that even *semantic* $\mathcal{P}$-refutations can be translated to *syntactic* resolution refutations. On the other hand, it would have been nice if the bound in part 4 of Lemma C.3 could have been made into a true upper bound in terms of the length of π_f, but it is easy to see that this is *not* possible. The reason for this is precisely that the $\mathcal{P}$-proof refuting $F[f]$ is allowed to use any arbitrarily strong semantic inference rules, and this can lead to exponential savings compared to syntactic resolution. For a concrete example, just let F be an encoding of the pigeonhole principle and let π_f be the refutation that downloads all axioms of $F[f]$ and then derives contradiction in one step.

Before proving Lemma C.3 let us see how it can be used to prove trade-offs provided that we can construct space-faithful projections.

Theorem C.4. *Let $\mathcal{P}$ be an implicational sequential proof system with space measure $Sp(\cdot)$. Suppose $f : \{0,1\}^d \mapsto \{0,1\}$ is a Boolean function such that there exists an f-projection which is space-faithful of degree K with respect to $\mathcal{P}$. Then if F is any unsatisfiable CNF formula and π_f is any semantic $\mathcal{P}$-refutation of the substitution formula $F[f]$, there is a resolution refutation π of F such that:*

- *The length of π is upper-bounded by π_f in the sense that π makes at most as many axiom downloads as π_f.*
- *The space of π is upper-bounded by π_f in the sense that $VarSp(\pi) = \mathrm{O}\big(Sp(\pi_f)^K\big)$.*

In particular, if there is no syntactic resolution refutation of F in simultaneous length $\mathrm{O}(L)$ and variable space $\mathrm{O}(s)$, then there is no semantic $\mathcal{P}$-refutation of $F[f]$ in simultaneous length $\mathrm{O}(L)$ and $\mathcal{P}$-space $\mathrm{O}\big(\sqrt[K]{s}\big)$.

Proof of Theorem C.4. Let π_f be a semantic $\mathcal{P}$-refutation of $F[f]$, and let π be the resolution refutation we obtain by applying the the projection $proj_f$ on π_f as in Lemma C.3. By part 4 of Lemma C.3 we know that π makes at most as many axiom downloads as π_f. By part 3 of the lemma we have $VarSp(\pi) = \mathrm{O}\big(\max_{\mathbb{D}\in\pi_f}\{VarSp(proj_f(\mathbb{D}))\}\big)$. Fix some $\mathcal{P}$-configuration $\mathbb{D}$ maximizing the right-hand side of this expression. For this $\mathbb{D}$ we have $VarSp(proj_f(\mathbb{D})) = \mathrm{O}\big(Sp(\mathbb{D})^K\big) = \mathrm{O}\big(Sp(\pi_f)^K\big)$ according to Definition 2.3. The theorem follows. □

Clearly, the key to the proof of Theorem C.4 is the claim that projections translate $\mathcal{P}$-refutations to resolution refutations. Let us substantiate this claim.

Proof of Lemma C.3. Fix any sequential proof system $\mathcal{P}$, any f-projection $proj_f$, and any CNF formula F. Recall that we want to show that if $\pi_f = \{\mathbb{D}_0, \mathbb{D}_1, \dots, \mathbb{D}_\tau\}$ is a semantic $\mathcal{P}$-refutation of the substitution formula $F[f]$, then the sequence of projected clause sets $\big\{proj_f(\mathbb{D}_0), proj_f(\mathbb{D}_1), \dots, proj_f(\mathbb{D}_\tau)\big\}$ is essentially a resolution refutation π except for some details that we might have to fill in when going from $proj_f(\mathbb{D}_{t-1})$ to $proj_f(\mathbb{D}_t)$ in the derivation.

Parts 1 and 2 of Lemma C.3 are immediate from Definition 2.2, since we have $proj_f(\mathbb{D}_0) = proj_f(\emptyset) = \emptyset$ by nontriviality and $\bot \in proj_f(\mathbb{D}_\tau)$ by completeness (note that $\mathbb{D}_\tau \vDash \bot = \bigvee_{x^b\in\bot} f^b(\vec{x})$ and the empty clause clearly cannot be derived by weakening).

We want to show that a resolution refutation of F can get from $proj_f(\mathbb{D}_{t-1})$ to $proj_f(\mathbb{D}_t)$ as claimed in part 3 of the lemma. For brevity, let us write $\mathbb{C}_i = proj_f(\mathbb{D}_i)$ for all i, and consider the possible derivation steps at time t.

Inference Suppose $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L_t\}$ for some L_t inferred from $\mathbb{D}_{t-1}$. Since $\mathcal{P}$ is an implicational proof system, it holds that $\mathbb{D}_{t-1} \vDash \mathbb{D}_t$, and since the projection is monotone by definition we can conclude that all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ are derivable from $\mathbb{C}_{t-1}$ by weakening. We go from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$ in three steps. First, we erase all clauses $C \in \mathbb{C}_{t-1}$ for which there are no clauses $C' \in \mathbb{C}_t$ such that $C \subseteq C'$. Then, we derive all

clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ by weakening, noting that all clauses needed for weakening steps are still in the configuration. Finally, we erase the rest of $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$. At all times during this transition from $\mathbb{C}_{t-1}$ to $\mathbb{C}_{t-1}$, the variable space of the intermediate clause configurations is upper-bounded by $\max\{ \mathit{VarSp}(\mathbb{C}_{t-1}), \mathit{VarSp}(\mathbb{C}_t)\}$.

Erasure Suppose $\mathbb{D}_t = \mathbb{D}_{t-1} \setminus \{L_{t-1}\}$ for some $L_{t-1} \in \mathbb{D}_{t-1}$. Again we have that $\mathbb{D}_{t-1} \vDash \mathbb{D}_t$, and we can appeal to the monotonicity of the projection and proceed exactly as in the case of an inference above.

Axiom download So far, the only derivation rules used in the resolution refutation π that we are constructing are weakening and erasure, which clearly does not help π to make much progress towards proving a contradiction. Also, the only properties of the f-projection that we have used are completeness, nontriviality, and monotonicity. Note, however, that a "projection" that sends $\emptyset$ to $\emptyset$ and all other configurations to $\{\bot\}$ also satisfies these conditions. Hence, the axiom downloads are where we must expect the action to take place, and we can also expect that we will have to make crucial use of the incremental soundness of the projection.

Assume that $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L_A\}$ for a function L_A encoding some clause from the substitution clause set $A[f]$ corresponding to an axiom $A \in F$. We want to show that all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived in π by downloading A, resolving (and possibly weakening) clauses, and then perhaps erasing A, and that all this can be done without the variable space exceeding $\mathit{VarSp}(\mathbb{C}_{t-1} \cup \mathbb{C}_t) \leq \mathit{VarSp}(\mathbb{C}_{t-1}) + \mathit{VarSp}(\mathbb{C}_t)$.

We already know how to derive clauses by weakening, so consider a clause $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ that cannot be derived by weakening from $\mathbb{C}_{t-1}$. By the incremental soundness of the projection, it holds for all literals $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ that the clauses $\overline{a} \vee C$ can be derived from $\mathbb{C}_{t-1}$ by weakening. Once we have these clauses, we can resolve them one by one with A to derive C.

Some care is needed, though, to argue that we can stay within the variable space bound $\mathit{VarSp}(\mathbb{C}_{t-1}) + \mathit{VarSp}(\mathbb{C}_t)$. Observe that what was just said implies that for all $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ there are clauses $\overline{a} \vee C_a \in \mathbb{C}_{t-1}$ with $C_a \subseteq C$. In particular, we have $\overline{a} \in \mathit{Lit}(\mathbb{C}_{t-1})$ for all $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$. This is so since by the incremental soundness there must exist some clause $C' \in \mathbb{C}_{t-1}$ such that $\overline{a} \vee C$ is derivable by weakening from C', and if $\overline{a} \notin \mathit{Lit}(C')$ we would have that C is derivable by weakening from C' as well, contrary to assumption. Note furthermore that if the projection is local, then $\overline{a} \vee C_a \in \mathbb{C}_{t-1}$ implies that $\overline{a} \vee C_a \in \mathbb{C}_t$ as well, since no clauses can disappear from the projection when enlarging $\mathbb{D}_{t-1}$ to $\mathbb{D}_t$. Thus, for local projections we have $\mathit{VarSp}(\mathbb{C}_{t-1} \cup \{A\}) \subseteq \mathit{VarSp}(\mathbb{C}_t)$.

If it happens that all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived by weakening, we act as in the cases of inference and erasure above. Otherwise, to make the transition from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$ in a space-efficient fashion we proceed as follows.

1. Erase all clauses in $\mathbb{C}_{t-1} \setminus \mathbb{C}_t$ not used in any of the steps below.
2. Infer all clauses in $\mathbb{C}_t \setminus \mathbb{C}_{t-1}$ that can be derived by weakening from $\mathbb{C}_{t-1}$.
3. Erase all clauses in $\mathbb{C}_{t-1} \setminus \mathbb{C}_t$ used in these weakening moves but not used in any further steps below.
4. Download the axiom clause A, and derive any clauses $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ such that $A \subseteq C$ by weakening.
5. For all remaining clauses $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ that have not yet been derived, derive $\overline{a} \vee C$ for all literals $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ and resolve these clauses with A to obtain C.
6. Erase all remaining clauses in the current configuration that are not present in $\mathbb{C}_t$, possibly including A.

Clearly, step 1 can only decrease the variable space, and steps 2 and 3 do not increase it. Step 4 can increase the space, but as was argued above we have $\mathit{Vars}(A) \subseteq \mathit{Vars}(C) \cup \mathit{Vars}(\mathbb{C}_{t-1})$ for every new clause C

derived with the help of A. Step 5 does not change the variable space, and step 6 can only decrease it. It follows that the set of variables mentioned during these intermediate steps is contained in $Vars(\mathbb{C}_{t-1} \cup \mathbb{C}_t)$. If in addition the projection is local, we have $\mathbb{C}_{t-1} \subseteq \mathbb{C}_t$ and also $Vars(A) \subseteq Vars(\mathbb{C}_t)$, so in this case the variable space increases monotonically from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$.

Wrapping up the proof, we have shown that no matter what $\mathcal{P}$-derivation step is made in the transition $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, we can perform the corresponding transition $\mathbb{C}_{t-1} \rightsquigarrow \mathbb{C}_t$ for our projected clause sets in resolution without the variable space going above $VarSp(\mathbb{C}_{t-1}) + VarSp(\mathbb{C}_t)$. Also, the only time we need to download an axiom $A \in F$ in our projected refutation π of F is when π_f downloads some axiom from $A[f_d]$. The lemma follows. □

C.2 Some Space-Faithful Projections

Let us pause and reflect on what Theorem C.4 says. Suppose we have a family of CNF formulas F_n with lower bounds for refutation variable space in resolution, or with trade-offs between refutation length and refutation variable space (such as for instance pebbling contradictions over suitable graphs). Then we can lift these lower bounds and trade-offs to stronger measures in a potentially stronger proof system $\mathcal{P}$, provided that we can find a Boolean function $f : \{0,1\}^d \mapsto \{0,1\}$ and an f-projection $proj_f$ that is space-faithful with respect to $\mathcal{P}$.

Thus, at this point we can in principle forget everything about proof complexity. If we want to prove space lower bounds or time-space trade-offs for a proof system $\mathcal{P}$, we can focus on studying Boolean functions of the form used by $\mathcal{P}$ and trying to devise space-faithful projections for such functions.

Let us now briefly review how this is done in [BN11], and then prove Lemma 2.4. In what follows, let us write $Vars^d(V) = \{x_1, \ldots, x_d \mid x \in V\}$ for the set of variables resulting from substitution in a formula over variable set V. Also, we will abuse notation mildly and identify a set of disjunctive clauses with the Boolean function computed by these clauses.

Definition C.5 (Precise implication [BN11]). Let f be a Boolean function of arity d, let $\mathbb{D}$ be a set of Boolean functions over $Vars^d(V)$, and let C be a disjunctive clause over V. If

$$\mathbb{D} \models C[f] \tag{C.1a}$$

but for all strict subclauses $C' \subsetneq C$ it holds that

$$\mathbb{D} \not\models C'[f] \ , \tag{C.1b}$$

we say that the clause set $\mathbb{D}$ implies $C[f]$ *precisely* and write

$$\mathbb{D} \rhd C[f] \ . \tag{C.2}$$

Definition C.6 (Resolution projection [BN11]). Let f denote a Boolean function of arity d and let $\mathbb{D}$ be any set of Boolean functions over $Vars^d(V)$. Then we define

$$Rproj_f(\mathbb{D}) = \{C \mid \mathbb{D} \rhd C[f]\} \tag{C.3}$$

to be the *resolution projection* of $\mathbb{D}$. Also, we define $Rproj_f^*(\mathbb{D}) = \bigcup_{\mathbb{D}' \subseteq \mathbb{D}} Rproj_f(\mathbb{D}')$ to be the *local resolution projection* of $\mathbb{D}$.

Lemma C.7. *The mapping $Rproj_f$ is an f-projection (for any sequential proof system $\mathcal{P}$).*

Proof. Suppose $\mathbb{D} \models C[f]$. Then we can remove literals from C one by one until we have some minimal clause $C' \subseteq C$ such that no more literal can be removed if the implication is to hold, and this clause C' is projected by $\mathbb{D}$ according to the definition. This proves both completeness and monotonicity for $Rproj_f$. Nontriviality is obvious.

For the incremental soundness, if $C \in Rproj_f(\mathbb{D} \cup \{L_A\})$ for an encoding L_A of some clause in $A[f]$, then this means, in particular, that $\mathbb{D} \cup \{L_A\} \models C[f]$. Consider any truth value assignment α such that $\alpha(\mathbb{D}) = 1$ but $\alpha\big(C[f]\big) = 0$. By assumption, $\alpha(L_A) = 0$. But this means that for all literals $a \in Lit(A)$ we have $\alpha\big(\overline{a}[f]\big) = 1$. Since this holds for any α, it follows for all $a \in Lit(A)$ that $\mathbb{D} \models (\overline{a} \vee C)[f]$, and we conclude by the completeness of the projection that the clause $\overline{a} \vee C$ is derivable by weakening from $Rproj_f(\mathbb{D} \cup \{L_A\})$. □

With this projection, and using Theorem C.4, the main technical result in [BN11] can now be rephrased as follows, where we also need the following key definition.

Definition C.8 (Non-authoritarian function [BN11]). We say that a Boolean function $f(x_1, \ldots, x_d)$ is *k-non-authoritarian*[13] if no restriction to $\{x_1, \ldots, x_d\}$ of size k can fix the value of f. In other words, for every restriction ρ to $\{x_1, \ldots, x_d\}$ with $|\rho| \leq k$ there exist two assignments $\alpha_0, \alpha_1 \supset \rho$ such that $f(\alpha_0) = 0$ and $f(\alpha_1) = 1$. If this does not hold, f is *k-authoritarian*. A 1-(non-)authoritarian function is called just *(non-)authoritarian*.

Theorem C.9 ([BN11]). *If f is a non-authoritarian Boolean function, then the projection $Rproj_f^*$ is exactly space-faithful with respect to the resolution proof system.*

We note in passing that [BN11] also proved a a similar result, although with slightly worse parameters, for the so-called k-DNF resolution proof systems (provided that the substitution function is $(k+1)$-non-authoritarian).

Lemma C.10 (Detailed version of Lemma 2.4). *If $\mathcal{P}$ is any implicational sequence proof system, then the projections $Rproj_f$ and $Rproj_f^*$ are both exactly space-faithful for variable space in $\mathcal{P}$.*

Proof. This is fairly straightforward. Consider first $Rproj_f$. If $Rproj_f$ is *not* exactly space-faithful, then there is a $\mathcal{P}$-configuration $\mathbb{D}$, a clause C, and a variable x such that $C \in Rproj_f(\mathbb{D})$ and $x \in Vars(C) \setminus Vars(\mathbb{D})$. Suppose without loss of generality that $C = C' \vee x$. By condition C.1b in Definition C.5 we have $\mathbb{D} \not\models C'$, so there is a truth value assignment α such that $\alpha(\mathbb{D}) = 1$ and $\alpha(C') = 0$. But we can flip α on x without affecting $\mathbb{D}$, since this variable does not occur, obtaining α' such that $\alpha'(\mathbb{D}) = 1$ and $\alpha'(C) = 0$. Contradiction.

For $Rproj_f^*$, we just focus on the subset $\mathbb{D}' \subseteq \mathbb{D}$ that is responsible for projecting C and then run the same argument. □

C.3 Designing Space-Faithful Projections for Stronger Proof Systems?

It seems like a very tempting approach to try to extend this projection framework to other proof systems than resolution and k-DNF resolution. In particular, it would be very interesting to see if one could prove space lower bounds and time-space trade-offs for cutting planes, polynomial calculus, or polynomial calculus resolution in this way. However, to do so another projection in the formal sense of Definition 2.2 than the one in Definition C.6 would be needed. We conclude this section by explaining why the same projection as we used for resolution above will *not* work for PC or PCR. Namely, consider the following examples (where the original variables before substitution are $x[i]$ for $i = 1, 2, \ldots$).

[13] Such functions have previously also been referred to as $(k+1)$-*robust* functions in [ABRW02].

Example C.11. Using substitutions with $\oplus_2$, for polynomial calculus we have the example

$$-1+\prod_{i=1}^{k} x[i]_1 x[i]_2 \tag{C.4}$$

showing that just two monomials can project the arbitrarily large conjunction $\overline{x[1]} \wedge \overline{x[2]} \wedge \cdots \wedge \overline{x[k]}$ if we use the projection in Definition C.6.

Example C.12. Let us also give a slightly more involved example for polynomial calculus resolution. For PCR, three monomials

$$-1+\prod_{i=1}^{k} x[i]_1 x[i]'_2 + \prod_{i=1}^{k} x[i]'_1 x[i]_2 \tag{C.5}$$

can project the arbitrarily large conjunction $x[1] \wedge x[2] \wedge \cdots \wedge x[k]$.

There are also similiar counter-examples that show why this projection does not work for cutting planes.

Somehow, the reason for these counterexamples is that the projection in Definition C.6 allows the Boolean functions in the implication to be far too strong. These functions do not really imply just conjunctions of exclusive ors, but something much stronger in that they actually fix the variable assignments (to some particular assignment that happens to satisfy exclusive ors). Note that formulas $F[\oplus_2]$ do not speak about fixed variable assignments for, say, x_1 or x_2, but only about the value of $x_1 \oplus x_2$. Intuitively, therefore, the only way we can know something more about x_1 and x_2 than the value of $x_1 \oplus x_2$ is if the refutation has already derived contradiction and is now deriving all kinds of other interesting consequences from this. But before this happens, we would like to argue that any refutation must pass through a stage where all it can know about x_1 and x_2 is the value of $x_1 \oplus x_2$ and nothing more.

For this reason, we would like to find a more "fine-grained" definition of a projection that can capture only these weaker implications and discard too strong implications. It seems like a very interesting, though also quite challenging, question whether it is possible to prove space lower bounds and/or trade-offs between proof length/size and space for cutting planes, polynomial calculus, or polynomial calculus resolution by designing smarter projections than in Definition C.6 that are space-faithful for these proof systems. And note that Definition 2.2 provides quite a lot of flexibility here—what we have proven in this appendix is that any function satisfying the formal conditions in Definition 2.2 will automatically project $\mathcal{P}$-refutations to resolution refutations for any (implications sequential) proof system $\mathcal{P}$.

D Trade-offs Between Space and Degree in PCR

In this short appendix, we show how Lemma C.10 can be used to prove Theorem 1.1, which we restate here for reference.

Theorem 1.1. *There is a family of 3-CNF formulas F_n of size $\Theta(n)$ that can be refuted in polynomial calculus in degree $Deg_{\mathcal{PC}}(F_n \vdash \bot) = \mathrm{O}(1)$ and also in monomial space $Sp_{\mathcal{PC}}(F_n \vdash \bot) = \mathrm{O}(1)$, but such that for any PCR-refutation $\pi_n : F_n \vdash \bot$ it holds that $Sp(\pi_n) \cdot Deg(\pi_n) = \Omega(n/\log n)$.*

We start by recalling some facts about the trade-off between clause space and width in resolution, obtained by studying a particular kind of pebbling contradictions.

Theorem D.1 ([Ben09]). *There is a family of k-CNF formulas F_n of size $\Theta(n)$ such that $Sp(F_n \vdash \bot) = \mathrm{O}(1)$ and $W(F_n \vdash \bot) = \mathrm{O}(1)$ but $VarSp(F_n \vdash \bot) = \Omega(n/\log n)$.*

We need a bit more information about these formulas as stated next.

Observation D.2 ([BIW04]). *There are resolution refutations $\pi_n : F_n \vdash \bot$ of the formulas F_n in Theorem D.1 with $L(\pi_n) = \mathrm{O}(n)$ and $W(\pi_n) = \mathrm{O}(1)$.*

Lemma D.3 ([Ben09]). *There are resolution refutations $\pi_n : F_n \vdash \bot$ of the formulas F_n in Theorem D.1 with $L(\pi_n) = \mathrm{O}(n)$ and $Sp(\pi_n) = \mathrm{O}(1)$, and all clauses in π_n contain at most one positive literal.*

Since the resolution refutation in Observation D.2 has constant width, polynomial calculus can simulate it in constant total degree. The resolution refutation in Lemma D.3 is wide, which could be a problem for PC, but only if there are clauses containing many positive literals (by our choice of encoding of true and false in PC). Since every clause in the refutation in Lemma D.3 contains at most one positive literal, PC can simulate this refutation as well with at most a (small) constant factor blow-up in the monomial space as compared to the resolution clause space.

The following observation brings us to a point where we can conclude the proof.

Observation D.4. *For any PCR-refutation π of a formula F it holds that $Sp(\pi) \cdot Deg(\pi) \geq VarSp(\pi) \geq VarSp_{\mathcal{PCR}}(F \vdash \bot)$.*

Proof. The refutation π never has more that $Sp(\pi)$ monomials in memory, and each monomial has degree at most $Deg(\pi)$. Thus the total number of distinct variables in memory at any point during the course of the PCR-refutation π is at most $Sp(\pi) \cdot Deg(\pi)$. □

And we clinch the argument by observing that the refutation variable space must be the same in PCR as in resolution.

Observation D.5. *For any CNF formula F it holds that $VarSp_{\mathcal{PCR}}(F \vdash \bot) \geq VarSp_{\mathcal{R}}(F \vdash \bot)$.*

Proof. Apply Lemma C.10 on $Rproj_f^*$ and then appeal to part 3 of Lemma C.3. □

Putting all of this together, Theorem 1.1 follows.

E Time-Space Trade-offs for PCR in the Sublinear Space Regime

Using the substitution theorem for PCR that we proved in Section 2.3, we can now go over the time-space trade-off results for resolution in [BN11] and lift most of them to PCR. The upper bounds will hold for total space syntactic resolution and polynomial calculus, whereas the lower bounds hold for monomial space in semantic PCR, i.e., a stronger space measure in a much stronger proof system. In contrast to [BN11], however, the upper and lower bounds in the trade-offs are no longer as tight, since the random restriction part of the argument leads to a loss of a logarithmic factor in the upper bound on the proof size. Let us recall our substitution theorem here for reference.

Theorem 2.5 (restated). *Suppose that F is a CNF formula for which any syntactic resolution refutation in variable space at most s must make more than T axiom downloads. Then any semantic PCR-refutation of $F[\oplus]$ in monomial space at most $s/\log_{4/3} T$ must have size larger than T.*

Given this theorem, there is a (literally) infinite supply of pebbling contradictions that it could be applied to in order to yield PCR time-space trade-offs. As was done also for the resolution trade-offs in [BN11], we try to simply state a few interesting examples here.

Theorem E.1 (Quadratic trade-offs for constant space). *There are explicit 6-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that the following holds:*

- *The formulas F_n are refutable in syntactic resolution and polynomial calculus in total space $\mathrm{O}(1)$.*

- *For any* $g(n) = \mathrm{O}(\sqrt{n})$ *there are syntactic resolution and polynomial calculus refutations* π_n *of* F_n *in simultaneous length/size* $\mathrm{O}\big((n/g(n))^2\big)$ *and total space* $\mathrm{O}(g(n))$.
- *For any semantic PCR-refutation* $\pi_n : F_n \vdash \bot$ *in monomial space* $Sp(\pi_n) \leq g(n)$ *it holds that* $S(\pi_n) = \Omega\left(\big(n/(g(n)\log n)\big)^2\right)$.

Theorem E.1 follows by combining Theorem 2.5 with the seminal work on pebbling trade-offs by Lengauer and Tarjan [LT82] and the structural results on simulations of black-white pebblings by resolution in [Nor12b].

Our next result relies on a new pebbling time-space trade-off result in [Nor12b], building on earlier work [CS80, CS82], which yields the rather striking statement that for any *arbitrarily slowly growing* non-constant function, there are explicit formulas of such (arbitrarily small) space complexity that nevertheless exhibit *superpolynomial* length-space trade-offs.

Theorem E.2 (Superpolynomial trade-offs for arbitrarily slowly growing space (detailed version of Theorem 1.2)). *Let* $g(n) = \omega(1)$, $g(n) = \mathrm{O}\big(n^{1/7}\big)$, *be any arbitrarily slowly growing function and fix any* $\varepsilon > 0$. *Then there are explicit* 6*-CNF formulas* $\{F_n\}_{n=1}^{\infty}$ *of size* $\Theta(n)$ *such that the following holds:*

- *The formulas* F_n *are refutable in syntactic resolution and PC in total space* $\mathrm{O}(g(n))$.
- *There are resolution and PC-refutations of* F_n *in simultaneous length/size* $\mathrm{O}(n)$ *and total space* $\mathrm{O}\left(\big(n/g(n)^2\big)^{1/3}\right)$.
- *Any PCR-refutation of* F_n *in monomial space* $\mathrm{O}\left(\big(n/(g(n)^3\log n)\big)^{1/3-\varepsilon}\right)$ *must have superpolynomial size.*

All multiplicative constants hidden in the asymptotic notation depend only on ε.

The two theorems above focus on trade-offs for formulas of low space complexity, and the lower bounds on length obtained in the trade-offs are somewhat weak—the superpolynomial growth in Theorem E.2 is something like $n^{g(n)}$. We next present a theorem that has both a stronger superpolynomial length lower bounds than Theorem E.2 and an even more robust trade-off covering a wider (although non-overlapping) space interval. This theorem again follows by applying our tools to the pebbling trade-offs in [LT82].

Theorem E.3 (Robust superpolynomial trade-off for medium-range space). *There are explicit* 6*-CNF formulas* $\{F_n\}_{n=1}^{\infty}$ *of size* $\Theta(n)$ *such that the following holds:*

- *The formulas* F_n *are refutable in syntactic resolution and PC in total space* $\mathrm{O}(\log^2 n)$.
- *There are syntactic resolution and PC-refutations of* F_n *in simultaneous length in length* $\mathrm{O}(n)$ *and total space* $\mathrm{O}(n/\log n)$.
- *Any semantic PCR-refutation of* F_n *in monomial space* $\mathrm{o}(n/\log^3 n)$ *must have size* $n^{\Omega(\log\log n)}$.

Having presented trade-off results in the low-space and medium-space range, we conclude by presenting a result at the other end of the space spectrum. Namely, appealing one more time to [Nor12b], we can show that there are formulas of polynomial (but still sublinear) space complexity that exhibit not only superpolynomial but even exponential trade-offs.

Theorem E.4 (Exponential trade-off (detailed version of Theorem 1.3)). *There is a family of explicit* 6*-CNF formulas* $\{F_n\}_{n=1}^{\infty}$ *of size* $\Theta(n)$ *such that the following holds:*

1. *The formulas F_n are refutable in syntactic resolution and PC in total space $\mathrm{O}\big(n^{1/11}\big)$.*

2. *There are syntactic resolution and PC-refutations π_n of F_n in simultaneous length/size $\mathrm{O}(n)$ and total space $\mathrm{O}\big(n^{3/11}\big)$.*

3. *Any semantic PCR-refutation of F_n in monomial space at most $n^{2/11}/(10\log n)$ must have size at least $\big(n^{1/11}\big)!$.*

As the knowledgeable reader might have noticed, however, we have no analogues of the strongest trade-offs in [BN11]. This is so because when the proof length in resolution becomes exponential, the log factor loss in the restriction argument becomes so big that it more or less cancels out the space lower bound.

F Extended Isoperimetry

For any undirected graph $G = (V, E)$, and $S \subseteq V$, let $\delta(S)$ denote the set of *boundary edges*, $\delta(S) := \{(v, v') \in E : \text{exactly one of } v, v' \in S\}$. An isoperimetric inequality in graph theory refers to a lower bound on $\delta(S)$ as above which holds for any subset of S in G depending only on the cardinality of S, and they have a long history of study. Our lower bound result refers to a variation on this idea, in which multiple sets of very different sizes are considered.

Definition F.1. Let $G = (V, E)$ be a undirected graph, and t_0 an associated parameter. A set of vertices of size between $t_0, |V|/2$ is called *medium sized*.

We say G satisfies the *extended isoperimetry* condition with parameters (W, t_0, r) if for any sequence of medium sized sets of vertices $S_1, \ldots, S_k \subseteq V$, where $\forall i \geq 2$, $|S_{i+1}| \geq r|S_i|$, it holds that $|\bigcup_i \delta(S_i)| \geq k \cdot W$.

In the sequel, we show that the grid has this property for some value of the parameters, which will be used for explicitly constructing Tseitin formulas not possible to be refuted within simultaneously small space and small size.

As a starting point, we first show that all path graphs have the property. It's easiest to start with the infinite path graph.

Let G_Z denote the graph defined by $V[G] := Z$, $E[G] := \{(i, i+1) : i \in Z\}$, that is, the undirected cayley graph on the integers with generator 1.

Lemma F.2. *Let $S_1, \ldots, S_k$ be a nonempty sequence of finite nonempty subsets of the integers such that $|S_1|, \ldots, |S_k|$ is a superincreasing sequence. (That is, each successive value is at least twice as large as the previous.) Then, $|\bigcup_i \delta(S_i)| \geq k + c$, where boundary refers to the graph G_Z and c is the number of connected components of the subgraph induced by $\bigcup_i S_i$ in G_Z.*

Proof. Let $H_{k,c}$ denote the proposition for specific values of k, c. We prove that $H_{k,c}$ holds for all values by induction on k and c. The case that $k = 1, c = 1$ is trivial.

Suppose inductively that $H_{k,c}$ holds. We prove it for $H_{k,c+1}$. For any sets $S_1, \ldots, S_k$ such that $H_{k,c+1}$ applies $\bigcup_i S_i$ has at least two connected components, so there exists a point z not in any set which is between two components and such that $z - 1$ is in one of the sets. Consider the function $f : Z \to Z$ defined by

$$f(x) := \begin{cases} x & x < z \\ x - 1 & x \geq z \end{cases} .$$

That is, f contracts the points $z, z + 1$ to one point. If $z + 1$ is not in any of the sets S_i, then f clearly preserves the size of every set, the number of boundary edges, and the number of connected components.

So in this case, $H_{k,c+1}$ holds for $S_1 \dots S_k$ if and only if it holds for $f(S_1), \dots, f(S_k)$. Since there is a point in one of the S_i which is above z, after a finite number of applications of this we obtain an instance such that $z+1$ is in one of the sets. Now we handle this case. Since z is not a point in any set, we still have $|f(S_i)| = |S_i|$ for every set. Since there is a component of $\bigcup_i S_i$ containing $z-1$ and component containing $z+1$, the images of these two components are one component afterwards, and no other collisions could have occurred so the number of components is reduced by exactly one. Further, there is at least one less edge in the $\bigcup_i \delta(f(S_i))$ compared with $\bigcup_i \delta(S_i)$ since the edges $(z-1, z)$ and $(z, z+1)$, which were boundary edges, collided after application of f, and no other edges collided or disappeared. By hypothesis that $H_{k,c}$ holds, we conclude that $|\bigcup_i \delta(f(S_i))| \leq k + c$, so $k + c + 1 \leq |\bigcup_i \delta(S_i)|$, so $H_{k,c+1}$ holds for this instance as desired.

Suppose inductively that $H_{k,c}$ holds for all c. We prove $H_{k+1,1}$. For any sets $S_1, \dots, S_{k+1}$ such that $H_{k,1}$ applies, let c' denote the number of connected components of $\bigcup_{i=1}^{k} S_i$. If $c' > 1$, then by $H_{k,c'}$, $|\bigcup_{i=1}^{k} \delta(S_i)| > k + 1$, so $|\bigcup_{i=1}^{k+1} \delta(S_i)|$ is at least as large and $H_{k+1,1}$ is satisfied. Suppose $c' = 1$. Then, the least and greatest points of $\bigcup_{i=1}^{k} S_i$ are at most $\sum_{i=1}^{k} |S_i| < |S_{k+1}|$ apart, so one of either the maximum or minimum point of S_{k+1} is an extreme point of $\bigcup_{i=1}^{k+1} S_i$ which is not in $\bigcup_{i=1}^{k} S_i$. Its boundary edge in the extreme direction is thus a boundary edge of S_{k+1} which is not a boundary edge of $\bigcup_{i=1}^{k} S_i$, which implies $|\bigcup_{i=1}^{k+1} \delta(S_i)| > |\bigcup_{i=1}^{k} \delta(S_i)| \geq k + 1$, so $H_{k+1,1}$ holds. □

Lemma F.3. *For any $\epsilon > 0$, for large enough n, the $n \times \ell$ grid, where $4n^2 \leq \ell \leq 2^n$, satisfies the extended isoperimetry condition with parameters $(W = n, t_0 = 4n^3, r = 2 + \epsilon)$.*

Proof. For a set of vertices S, we call a column of the grid *full* if all its n vertices are included in S, *empty* if none of its vertices is included, and *partial* otherwise. If for some S_i, there are more than kn partial columns, each of which introduces at least one vertical boundary edge, thus the lemma holds. Therefore without loss of generality, the number of partial columns for every S_i is less than $kn < n^2$ (note that $k < n$).

It suffices to show that in each row we obtain at least k horizontal boundary edges, since an edge is a horizontal edge in at most one row and there are n rows. Fix any row, and for each S_i let S_i' denote the index number of columns which contain a vertex of S_i' from this row. If we can show that all ratios $|S_i'|/|S_{i-1}'|$ are at least 2, then the previous lemma implies that $|\bigcup \delta_{G_Z}(S_i')| \geq k + c$, where c is the number of components of $\bigcup S_i'$ in G_Z. For every edge of this union *except* the edges $(0, 1)$ and $(\ell, \ell + 1)$ there is a corresponding horizontal edge of $\delta(S_i)$ in this row. If $|\bigcup \delta_{G_Z}(S_i')|$ contains at most one of these two, then since $c \geq 1$ we obtain at least k horizontal edges as desired. If $|\bigcup \delta_{G_Z}(S_i')|$ contains both of $(0, 1)$, $(\ell, \ell + 1)$, then $c \geq 2$. For suppose $c = 1$, then $1, \ell$ are in the same connected component of $\bigcup S_i'$, yet $|\bigcup S_i'|$ is strictly less than ℓ, since the seuence is superincreasing and hence $\sum_i |S_i'|$ is upper bounded by $2|S_k'|$. Thus in this case as well, there are at least k horizontal edges as desired.

Now we bound the ratios $|S_i'|/|S_{i-1}'|$.

$|S_i'| \geq |S_i|/n - n^2$, since $|S_i'|$ is at least the number of full columns in S_i; and also $|S_i'| \leq |S_i|/n + n^2$, since there are at most $|S_i|/n + n^2$ columns which are full or partial. By assumption, $|S_{i+1}| \geq r|S_i|$. Therefore,

$$|S_{i+1}'|/|S_i'| \geq \frac{|S_{i+1}|/n - n^2}{|S_i|/n + n^2} \geq r(1 - n^2/t_0)^2 = r(1 - o(1)), \forall i : 1 \leq i \leq k' - 1$$

here using that $t_0 = 4n^3$ is a lower bound on $|S_i|$.

Thus for any $\epsilon > 0$, for large enough n, $r(1 - o(1))$ will exceed two as required. This completes the proof. □

G Trade-offs For Tseitin Tautologies

G.1 Overview

A long line of work [CEI96, IPS99, BW01] has resulted in several techniques in Resolution for proving a Size lower bound for proofs of a tautology from a Width lower bound for proofs of that tautology – here, width refers to the maximum over all clauses in the proof of the number of variables in the clause. Proving lower bounds on proof width is significantly simpler, and many techniques exist which we will discuss later. These techniques take several forms – in the very well known work of Ben-Sasson & Wigderson [BW01], a very strong width lower bound for a formula is shown to immediately imply a very strong size lower bound for that formula. However, more relevant to us is in an older work of Beame & Pitassi [BP96], showing that if a formula has a nice distribution of random restrictions associated to it such that the restricted formula satisfies a width lower bound of any kind, then that formula obtains a corresponding size lower bound. This argument often yields essentially tight size lower bounds at many different ranges of hardness, in contrast to the technique of Ben-Sasson & Wigderson. In [Ben09], a generic technique was given for taking a formula F and obtaining a formula F' which has such a nice distribution of restrictions, all of which yield F. This technique is $\oplus$-substitution which we defined earlier. In the case of Tseitin Tautologies which we study, this technique gives essentially optimal size lower bounds – that is, the easy and tight width lower bound for F coupled with this argument gives a size lower bound for $F[\oplus]$ which is tight up to a constant in the exponent.

In the next section, we will show that the results of [BBI12] can be cast in the following terms: If F satisfies an extended width lower bound property, then $F[\oplus]$ obeys a time space trade-off lower bound, which in the high space regime agrees with (and thus extends) the size lower bound just described. In the case of Tseitin on the grid, the size lower bound is optimal up to constant factors, so this yields a true time space trade-off result. To establish the extended width lower bound for Tseitin formulas on a grid, it is sufficient to augment existing techniques for width lower bounds with the extended isoperimetric property of the grid described in Appendix F. Our arguments will show that any graph with extended isoperimetry would do here. Additionally, this improves over the results of [BBI12] by permitting us to carry out the argument on k-CNFs for $k = O(1)$ rather than growing with the size of the formula.

In the subsequent section, we will discuss how to extend this paradigm to give lower bounds in PCR. In PCR a roughly analogous size vs. width paradigm exists, where degree is substituted for width. In [BGIP01], a beautiful technique for this was to take linear substitutions for the variables of a formula, and show a degree lower bound for proofs of the substituted formula. For Tseitin Tautologies, the most natural map to take is the Fourier Transform, and a series of other technical insights related to Binomial PC results in a very intuitive degree lower bound result for Fourier Transformed Tseitin and hence standard Tseitin. In many ways this result is a natural extension of the corresponding results for Resolution. We revisit these techniques and show that with additional careful analysis, they can be married with the ideas of Appendix G.5 to give essentially the same results for PCR as we obtained for Resolution in Appendix G.3.

G.2 Resolution Refutations of Tseitin Formulas

G.2.1 Measures of Progress

A standard technique described by [BW01] to show width lower bounds is to take advantage of the following measure of progress of any proof, which is definable in any proof system. Let A be a set of contradictory formulae. For ϕ a proof line in a refutation of A, let

$$\mu_A(\phi) := \min_{\substack{S \subseteq A \\ S \models \phi}} |S| .$$

That is, we ignore issues of whether ϕ can be feasibly proven from A and just consider how many of the axioms in A we need to *semantically* imply ϕ to judge roughly how valuable it is in the proof.

This simple measure enjoys nice properties. We always have $\mu_A(a) = 1$ for $a \in A$, and μ is always *subadditive*: If $\phi_1, \phi_2 \vdash \phi_3$ is a step of any proof in a sound proof system, then $\mu_A(\phi_1)+\mu_A(\phi_2) \geq \mu_A(\phi_3)$. This is because if the derivation is sound, then the union of the minimium sets of axioms for ϕ_1, ϕ_2 will semantically imply both ϕ_1 and ϕ_2, and thus ϕ_3. So for instance, if π is any refutation of A, there will always exist a "medium complexity" formula $\phi \in \pi$ such that $\mu_A(\bot)/3 \leq \mu(\phi) < 2\mu_A(\bot)/3$. For suppose there wasn't – then the first time a formula of complexity exceeding $2\mu_A(\bot)/3$ was derived in the refutation, it was derived from two formulae of complexity less than $\mu_A(\bot)/3$, contradicting subadditivity.

An important technical insight concerning such measures [BW01] is that if A is well chosen and the lines of the proof system we consider have restricted expressive power, then such a "medium complexity" ϕ will necessarily be complicated as a boolean formula. In the case of Resolution, ϕ will be a clause with many variables, and so in this way we obtain a width lower bound for refutations of A. Generally, it is sensible to choose A to be minimally contradictory, since this gives us $\mu_A(\bot) = |A|$, and for any contradictory A we can usually think of an interesting minimally contradictory subset. In the sequel we'll restrict attention to this case.

Several authors [PI00, BBI12] observed that in fact this argument shows much more; there is also a ϕ' with $\frac{\mu_A(\phi')}{\mu_A(\bot)} \in [1/6, 1/3)$, a ϕ'' with $\frac{\mu_A(\phi')}{\mu_A(\bot)} \in [1/12, 1/6)$, etc. While the existence of several collectively wide clauses can be useful for some applications, to obtain time space trade-offs we need to use more than this. For the sequel, its convenient to transition to the following definition.

Definition G.1. A function μ' mapping formulas of a proof system to $\mathbb{N}$ is a *strongly bounded* if, whenever $\phi_1, \phi_2 \vdash \phi_3$ follows in one proof step, then $\mu'(\phi_3) \leq 1 + \max(\mu'(\phi_1), \mu'(\phi_2))$.

Such a μ' is a a *strongly bounded complexity measure for* A if it is strongly bounded and $\mu'(a) = 0$ for all $a \in A$.

Of course, it's easy to see that if μ_A is a subadditive complexity measure as defined before, then $\mu'_A := \lfloor \log_2 \mu_A \rfloor$ is a strongly bounded complexity measure for A. For us, this measure μ' plays the role of "complexity levels" as appeared in [BBI'12]. We've chosen to abstract some of this detail in part because strongly bounded measures constructed in this fashion will always have $\mu'_A(\bot) \leq \log_2 |A|$, but its conceivable that for a special class of formulae such a measure could be constructed with a much larger value here, which corresponds to having many more complexity levels and a much stronger lower bound bound via our technique, provided it enjoys the other necessary properties as we shall see.

One of the key technical points in [BBI12] was to observe that, having such a complexity measure not only implies that a clause of every complexity value appears, but that they appear in a certain order and represent points on an unbroken chain in the proof. This means that for instance, if at the start of some period of time in the proof only low complexity formulas are in memory, and at the end there are high complexity formulas, then the intervening middle complexity clauses which we are guaranteed by subadditivity actually occurred within in this epoch. The following simple lemma is ultimately central to the time space trade-off proof.

Definition G.2. Let π be a proof in a sequential proof system which is divided recursively into epochs. For E an epoch, say that its *critical set* of proof lines is

- If E is an internal epoch, the set of proof lines appearing in memory at breakpoints between sons of E
- If E is a leaf epoch, the set of all proof lines appearing in E.

Lemma G.3 (Subdivision argument for progress levels). *Let A be a set of formulas from a sequential proof system $\mathcal{P}$ such that there is a strongly bounded complexity measure μ' for A, and $\mu'(\bot) = L$. If a refutation of A is divided recursively into epochs to a recursive depth of h, then one of the following holds.*

1. *There exists a leaf epoch containing $L \cdot k^{-h}$ formulae with distinct values under μ'.*
2. *There exists an internal epoch such that the critical set of proof lines contains at least k formulae with distinct values under μ'.*

Proof. Say that an internal epoch represents a "progress gap" of g if for some ℓ, h, $g = h - \ell$ every formula ϕ in the initial configuration has complexity at most ℓ, but the final configuration contains a formula of complexity at least h. By definition, the entire proof represents a progress gap of at least L.

Suppose that the second condition fails. Then for any internal epoch representing a gap of g, one of its children represents a gap of at least g/k; since only k complexities appear at breakpoints between children of this epoch, there exist by averaging a successive pair of these values ℓ', h' with $h' - \ell' > g/k$, and so the first time a value of complexity $\geq h'$ appears at such a breakpoint, the epoch immediately preceeding it begins with only formulae of complexity $\leq \ell'$ and so this epoch respresents a progress gap of g/k.

By induction, we conclude that some leaf epoch represents a progress gap of at least $L \cdot k^{-h}$; let ℓ, h be according to the definition. At the beginning of the epoch we have only formulas of complexty at most ℓ, but at the end we have a formula of complexity at least h, so by strong boundedness, there is also a formula in this epoch of complexity $h - 1$. Consider the first such formula. Again by strong boundedness, there is a formula of this epoch of complexity $h - 2$, and so on, repeating the argument until a formula of complexity $\ell + 1$ is found. Thus this leaf epoch contains formulas of at least $L \cdot k^{-h}$ complexities. This completes the proof. □

G.2.2 Width Lower Bounds from Progress Measures, Tseitin

For the class of Tseitin formulas, the Ben-Sasson Wigderson style complexity measure described is very well studied. A well known application is that if a graph has no balanced cut of size less than W, where balanced means between $1/3$ and $2/3$ of the vertices on either side, then the corresponding Tseitin formula has a clause of width at least W in any refutation. Follow up work [BI10, AR02] has shown that this is generally tight. The idea is to show that if C is a clause then every variable on the boundary of S_C appears in C. To carry out the [BBI12] strategy we need to prove this or something similar. We will also need a stronger version for when we ultimately handle polynomial calculus, so we'll state this lemma in slightly greater generality, which we can do without complicating the proof.

Lemma G.4. *Let ϕ be a disjunction of $\mathbb{F}_2$-linear equations in the variables x_e of a Tseitin formula on graph G. Let S be a minimal subset of the vertices such that $\{PARITY_v\}_{v \in S} \models \phi$. Then, $Vars(\phi) \supseteq \{x_e : e \in \delta(S)\}$.*

Proof. The assumptions imply that S is a minimal subset such that $\{PARITY_v\}_{v \in S} \wedge \neg\phi \models \bot$. Since ϕ is a disjunction of $\mathbb{F}_2$ equations, $\neg\phi$ is a conjunction of $\mathbb{F}_2$ equations, and our assumption is that $\{PARITY_v\}_{v \in S} \wedge \neg\phi$ is an inconsistent system of equations. Without loss of generality, we may assume that $\neg\phi$ is satisfiable, since if not then minimality of S implies that S is empty and the claim is vacuous.

By basic linear algebra this implies that it is possible to derive $0 = 1$ from the system via $\mathbb{F}_2$-linear combinations. Suppose that in this linear combination, one of the equations $PARITY_v$ has a coefficient of 0. Then, when we remove that equation, the linear combination would still derive $0 = 1$ from the subsystem, so that subsystem is inconsistent, contradicting minimality of S. Therefore each $PARITY_v$ equation has a coefficient of 1 in this linear combination, or in otherwords, the sum of the $PARITY_v$ equations are inconsistent with $\neg\phi$. If there is a variable of this sum which does not appear in $\neg\phi$, then clearly the sum

and $\neg\phi$ are satisfiable if $\neg\phi$ is. So we must have $Vars(\sum_{v\in S} PARITY_v) \subseteq Vars(\neg\phi)$. Now we use the structure of the Tseitin equations. Every variable x_e occurs in exactly two equations $PARITY_v$, and so if $e \in \delta(S)$, then x_e occurs in exactly one of the summands in $\sum_{v\in S} PARITY_v$. Therefore it does not cancel in the sum and is a variable of $\sum_{v\in S} PARITY_v$. This completes the proof. □

The previous lemma gives a connection between isoperimetry in a graph and the following standard complexity measure [BW01] for Tseitin.

Definition G.5 (Subadditive Complexity Measure for Tseitin). Let C be a clause in the variables of a Tseitin formula on a connected graph G. Then define the measure μ by

$$\mu(C) := \min_{\substack{S \subseteq V \\ \{PARITY_v\}_{v\in S} \models C}} |S| .$$

Further, let S_ϕ denote any fixed set S achieving the minimum above for ϕ.

Observation G.6. *1. For a any axiom in any representation of Tseitin, $\mu(a) = 1$.*

2. $\mu(\bot) = |V|$.

3. μ is subadditive.

4. $\{x_e : e \in \delta(S_C)\} \subseteq Vars(C)$.

Proof. The first three follow directly from the discussion. To see the fourth, observe that a clause can be thought of as a disjunction of $\mathbb{F}_2$-linear equations, so Lemma G.4 applies. □

G.3 Time Space Trade-off for Resolution

Now we're ready to show that extended isoperimetry implies a Time Space Trade-off.

Proposition G.7. *If a graph $G = (V, E)$ satisfies extended isoperimetry with parameters W, t_0, and μ' is the strong complexity measure defined from the previous complexity measure μ via*

$$\mu'(C) = \begin{cases} 0 & \mu(C) < t_0 \\ L & \mu(C) > |V|/2 \\ \log_2(\mu(C)/t_0) & \text{otherwise} \end{cases}$$

then for any k clauses $C_1, \ldots C_k$ with distinct values under μ' between 0 and L,

$$\left|\bigcup Vars(C_i)\right| \geq \Omega(kW) . \tag{G.1}$$

Proof. Order the C_i by value of μ'. If the C_i have distinct values under μ', then $\mu'(C_{i+3}) \geq \mu'(C_i) + 3$, and by definition of μ', $\mu(C_{i+3}) \geq 4 \cdot \mu_{C_i}$. This implies $|S_{C_{i+3}}| \geq 4 \cdot S_{C_i}$, and extended isoperimetry implies $|\bigcup_i \delta(S_{C_{3i+1}})| \geq kW/3$. By observation G.6, this implies $|\bigcup_i Vars(C_i)| \geq kW/3$ as desired.

□

Theorem G.8. *If F is any CNF with a strong complexity measure μ for F satisfying Equation G.3, and $\mu(\bot) = L$, then $F[\oplus]$ satisfies $(2^{\Omega(W)}/S)^{\Omega\left(\frac{\log\log L}{\log\log\log L}\right)}$.*

Proof. Let π be any proof of $F[\oplus]$. Divide π into epochs recursively to a recursive depth of h, dividing each epoch into m equal subepochs, h, m to be determined later. Now consider the random restriction ρ. Since $F[\oplus]\restriction_\rho = F$, $\pi\restriction_\rho$ is a refutation of F. Choose k such that $Lk^{-h} = k$. Since μ' is a strongly bounded complexity measure, Lemma G.3 implies that the critical set of some epoch of $\pi\restriction_\rho$ contains at least k clauses of distinct complexity values, with probability one.

However, for M any small set of clauses, the probability that $M\restriction_\rho$ contains k distinct complexities is seen to be small. If a collection of clause each is not restricted to a constant by ρ, then their disjunct is not restricted to a constant either. Therefore by Lemma B.11, the probability that any fixed k-tuple of clauses all survive and have collective width $\geq X$ is at most exponentially small in X. In our case, the collective width is at least $\Omega((k-2)\cdot W)$, by Proposition G.7. By a union bound over all k-tuples of M, the probability that any k of them have distinct complexity values is at most $M^k 2^{-\Omega((k-2)W)}$.

Choose the parameter m so that $mS = T/m^{h-1}$, so that all critical sets of epochs have the same size. Choose h so that $h = k$ and $k^h = L$. The probability that any critical set of any epoch contains k complexities after the restriction is at most $(mS\exp(-\Omega(W)))^k$, and there are at most m^h epochs. By a union bound, the probability that any critical set of any epoch contains k complexities after the restriction is at most $\left(m^2 S\exp(-\Omega(W))\right)^k$. Since we know this must happen with certainty, we conclude $m^2 \geq 2^{\Omega(W)}/S$, or $T \geq \left(2^{\Omega(W)}/S\right)^{\Omega(k)}$. Here k is such that $k^k = L$, so in terms of our L our exponent exceeds $\log L/\log\log L$, as desired. □

In Appendix F we gave a simple proof that the $n\times\ell$ grid satisfies extended isoperimetry with parameters $(W = n, t_0 = 4n^3, r = 2+\epsilon)$, so this theorem yields lower bounds of the form $\left(2^{\Omega(n)}/S\right)^{\Omega\left(\frac{\log\log n}{\log\log\log n}\right)}$. For a discussion of the relevant upper bounds on these formulas, see Section 2.5.

G.4 PCR Refutations of Tseitin Formulas

G.4.1 Overview

Our main result shows that actually, the trade-off lower bounds we just prove hold in PCR for these formulas. In the prevailing philosophy, Resolution Size lower bounds via Width lower bounds are analogous to PCR Size Lower bounds via Degree lower bounds, as we discussed. Thus, one would expect that an "extended degree lower bound" analogous to Equation G.3 would imply time space trade-offs via the subdivision and restriction argument we just saw. However, degree lower bounds for PCR are significantly more challenging to prove in general, and many of the techniques that have been developed (e.g. [AR]) appear ill-suited for obtaining an extended degree lower bound. We illustrate one elegant solution which builds on the techniques of [BGIP01] for degree lower bounds. The crux of the argument is a pair of simulations, which while in general not "efficient" as measured by most standard proof complexity measures, are efficient with respect to degree. We revisit these simulations and give a refined analysis. The simulations convert a PCR refutation to a Binomial PC refutation, which is in a sense a stone's throw away from a Resolution refutation. Using a standard Ben-Sasson Wigderson style complexity measure, we're able to extend degree lower bounds in this setting directly from Lemma B.11 and extended isoperimetry. The flavor of the final proof changes slightly from what we saw in Resolution – ultimately we don't establish a strongly bounded complexity measure for PCR with the extended degree lower bound, but we do establish such a measure for Binomial PC, and modulo the simulations, this is enough for the subdivide-and-restrict approach to work.

G.5 PCR Simulations

Essential to our argument is a pair of simulations connecting PCR refutations of standard Tseitin to refutations of a "fourier transformed" version of Tseitin. These simulations were exploited by [BGIP01] for

proving degree lower bounds, and are also essential for our time space trade-off lower bounds.

First, we introduce an alternate formulation of Tseitin as a system of Binomials over a field of odd characteristic.

G.5.1 Fourier-Transformed Tseitin

The previous (CNF) will be called $\{0,1\}$-Tseitin (suppressing G, χ for now). Following [BGIP01], we define an alternate instance $\{+1,-1\}$-Tseitin, formed by defining fourier transformed variables $y_e := 1 - 2x_e$, that is, $x_e = 0 \iff y_e = 1, x_e = 1 \iff y_e = -1$, and expressing the parity constraints in this new basis. Now, the value of a linear equation $\bigoplus x_e$ corresponds to the value of $\prod y_e$ under this correspondence, due to the fourier transform.

Definition G.9. Given G, χ, the PCR instance of $\{+1,-1\}$-Tseitin is defined by variables

$$\begin{aligned} \text{Variables: } \forall e \in E & \qquad y_e \\ \{\text{+1,-1}\}\text{-Constraints: } \forall e \in E & \qquad y_e^2 - 1 = 0 \\ \text{Parity Constraints: } \forall v \in V, & \qquad \prod_{e \sim_G v} y_e = (-1)^{\chi(v)} \end{aligned}$$

The primary reason why $\{+1,-1\}$-Tseitin is preferable is that every axiom above is a Binomial, which wasn't the case for $\{0,1\}$-Tseitin. Since binomial systems are much simpler the reduction we will see next is profitable to consider.

G.5.2 Reductions

Definition G.10. Let π be a PCR refutation of $\{0,1\}$-Tseitin. We define an induced / simulated PC refutation π' of $\{-1,1\}$-Tseitin in the straightforward way, mantaining the invariant that whenever π has a configuration $\mathcal{P}$, we will have a corresponding configuration with polynomials in the variables y_e which are semantically equivalent modulo $y_e = 1 - 2x_e$.

- When π downloads an axiom, π' downloads a semantically equivalent axiom in the new basis, by downloading one of the $\{-1,1\}$ axioms and weakening it.
- When π performs a weakening step, we perform a weakening and a linear combination step, and an erasure step. (when $p \vdash x_i \cdot p$ is inferred, we must simulate this with $p' \vdash y_i \cdot p'$ followed by $p', y_i \cdot p' \vdash (1 - 2y_i) \cdot p'$ to obtain a semantically equivalent polynomial)
- When π performs a linear combination step, we do the same to the corresponding polynomials.
- When π erases an polynomial, we erase the analogous polynomial.

The correctness of this simulation should clear, and it should be clear that when π is in fact a refutation, π' will also be a refutation.

For [BGIP01] the crucial property of this simulation was that it didn't increase the degree – since the "fourier transform" represents a linear substitution of the variables, if there was no monomial of large degree before the substitution won't introduce one. For us, the crucial property of this simulation is that it is "conservative with respect to monomials".

Claim G.11. The simulation of PCR on $\{0,1\}$-Tseitin by PC on $\{+1,-1\}$-Tseitin is *conservative with respect to monomials*;

- If for some *configuration* of the simulated proof no active monomial contains the set of variables $\{x_e : e \in E'\}$ for some $E' \subseteq E$, then the corresponding configuration of the simulating proof doesn't contain any active monomials containing all of $\{y_e : e \in E'\}$.

- If for some *time period* in the simulated proof no active monomial contains the set of variables $\{x_e : e \in E'\}$ for some $E' \subseteq E$, then the corresponding time period of the simulating proof doesn't contain any active monomials containing all of $\{y_e : e \in E'\}$.

Proof. The only steps which introduce new monomials are download and weakening steps. The only steps which cause monomials to dissappear from the original proof are erasure steps. It is trivial to see that none of these will cause of violation of conservativity. □

Our second simulation is more sophisticated; it requires a clever algebraic argument which will allow us to work with polynomials of a simple form. This lemma is an adaptation of the main technical lemma of [BGIP01].

Lemma G.12. *Suppose $\mathcal{B}$ is a set of binomials with coefficients in a field. Suppose p is a polynomial which can be written as a linear combination of elements of $\mathcal{B}$. Then by repeatedly taking linear combinations of binomials of $\mathcal{B}$ which each time yield a binomial and adding them to $\mathcal{B}$, it is possible to obtain a larger $\mathcal{B}'$ such that p is a no-cancellation linear combination of elements of $\mathcal{B}'$, i.e. for some coefficients α, $p = \sum_{b \in \mathcal{B}'} \alpha_b \cdot b$ and in this sum every monomial which appears in a summand appears in p.*

Proof. The proof is by induction on the number of cancelling monomials. We show that if p can be written as a sum with k cancelling monomials, then some number of steps can be performed for $\mathcal{B}$ to obtain $\mathcal{B}'$ such that p is now expressable as a sum with only $k-1$ cancelling monomials.

Let m denote some monomial which cancels.

Consider the subsum of the sum yielding p above obtained by selecting only the binomials which contain m. Because we are working over a field, this subsum may be expressed

$$\sum_{k=1}^{T} \beta_k (m - t_k) \ ,$$

where β_k are nonzero field coefficients and t_k is some term, by factoring out the coefficient on m from each binomial in the subsum. Since m cancels in this sum by assumption, $\sum \beta_b = 0$. Now, we claim that the subsum can be rewritten

$$\sum_{k=2}^{T} \beta_k (t_1 - t_k) \ .$$

This is because

$$\begin{aligned}
\sum_{k=2}^{T} \beta_k(t_1 - t_k) &= \left(\sum_{k=2}^{T} \beta_k\right) t_1 - \sum_{k=2}^{T} \beta_k t_k \\
&= -\beta_1 t_1 - \sum_{k=2}^{T} \beta_k t_k \\
&= -\sum_{k=1}^{T} \beta_k t_k \\
&= \left(\sum_{k=1}^{T} \beta_k m\right) - \sum_{k=1}^{T} \beta_k t_k \\
&= \sum_{k=1}^{T} \beta_k (m - t_k)
\end{aligned}$$

Clearly, $t_1 - t_k$ is a linear combination of $\beta_1(m - t_1)$ and $\beta_k(m - t_k)$. Thus, if we derive $t_1 - t_k$ for each $2 \leq k \leq T$ and add it to $\mathcal{B}$, the subsum we considered can also be derived as a linear combination of the additional binomials, without containing the monomial m in any summand. Now, in the original sum for p, replace the subsum we considered with this new sum. We obtain a new sum that yields p, but which never mentions m, and we could not have introduced any new monomials since all we did was take linear combinations of the old summands. Thus p is a linear combination with one fewer noncancelling monomial. This completes the proof. □

Definition G.13. Let π' be a PC refutation of $\{-1, 1\}$-Tseitin. We define an induced / simulated Binomial PC refutation π'' following the ideas of [BGIP01].

The simulation invariant is that whenever π' has a configuration $\mathcal{P}$, we will have a corresponding configuration in which every polynomial $p \in \mathcal{P}$ will be a no-cancellation linear combination of currently active binomials. That is, every polynomial p will be a linear combination of binomials from the simulating configuration, and no monomials will cancel in this sum. In the following, we will assume inductively that this is the case and fix such a linear combination for each p, and for each p the binomials in its linear combination will be termed the binomials *underlying* p. Note that this simulation *is* efficient, broadly speaking, but again this won't actually be important for our proof.

- When π' downloads an axiom, π'' downloads the same axiom, which is a binomial.
- When π' performs a weakening step on p to obtain result r, we perform the same weakening step for each underlying binomial. In the next configuration, these binomials underly r.
- When π' performs a linear combination step $p, q \vdash \alpha p + \beta q$ to obtain result r, we observe that r is now a linear combination of the binomials underlying p and q, with cancellation. By Lemma G.12, r is also a linear combination without cancellation of binomials which may be inferred by linear combinations from the set of binomials underlying p, q, so π'' performs these linear combination inferences, and in the next configuration, these binomials underly r.
- When π' performs an erasure step, π'' erases any binomials which no longer underly any polynomial.

Observation G.14. *If π' is a refutation, π'' is also a refutation.*

Proof. Suppose a configuration $\mathcal{P}$ of π' contains the contradition $1 = 0$. Then the corresponding Binomial configuration contains $c = 0$ for some nonzero field element c – for suppose not, then there cannot be a no-cancellation linear combination of its binomials yielding 1. □

Claim G.15. The BGIP simulation of PC on $\{+1, -1\}$-Tseitin by Binomial PC is conservative with respect to monomials;

- If for some *configuration* of the simulated proof no active monomial contains the set of variables $\{y_e : e \in E'\}$ for some $E' \subseteq E$, then the corresponding configuration of the simulating proof doesn't contain any active monomials containing all of $\{y_e : e \in E'\}$.
- If for some *time period* in the simulated proof no active monomial contains the set of variables $\{y_e : e \in E'\}$ for some $E' \subseteq E$, then the corresponding time period of the simulating proof doesn't contain any active monomials containing all of $\{y_e : e \in E'\}$.

Proof. Download steps will obviously never pose a problem. A single weakening step for the PC proof may correspond to many weakening steps for the Binomial proof, but no other varieties of steps occur, so it is straightforward to see that conservativity will hold here. Similarly, a single linear combination step for the PC proof introduces no new monomials, and is simulated by a series of linear combination steps in the Binomial proof, with no other varieties of steps occuring. So inference steps will not pose a problem.

Finally consider erasure steps. If after an erasure step in the simulated proof there is a monomial in some binomial of the simulating proof which does not appear in the simulated proof, then this binomial cannot be underlying any polynomial of the simulated proof, since in any no-cancellation sum this monomial would remain. Thus by the end of the simulating erasures, every monomial in the simulating proof appears in the simulated proof, so conservativity holds. □

G.5.3 Time Space Trade-off for PCR

As discussed in the overview, Binomial PC is simple enough that we can obtain a strongly bounded complexity measure with the extended degree lower bound by generalizing the standard complexity measure (definition above).

Definition G.16. Let $b = 0$ be a Binomial PC proof line in the variables of $\{+1, -1\}$-Tseitin on graph $G = (V, E)$. Define the binomial complexity measure

$$\mu(b = 0) := \min_{\substack{S \subseteq V \\ \forall e \in E, (x_e = 0 \wedge y_e = 1) \vee (x_e = 1 \wedge y_e = -1)\}, \{PARITY_v\}_{v \in S} \models b = 0}} |S| \,.$$

Further, let S_b denote any fixed set S achieving the minimum above.

The first three parts of Observation G.6 follow easily in this setting as well.

Observation G.17. *μ is a subadditive complexity measure in the sense of Ben-Sasson & Wigderson [BW01]. That is,*

- *For any axiom $b = 0$ of $\{+1, -1\}$-Tseitin, $\mu(b = 0) = 1$.*
- *$\mu(\bot) = |V|$.*
- *μ is subadditive.*

To obtain the last, we appeal to Lemma G.4.

Corollary G.18. *For any $b = 0$ Binomial PC proof line in the variables of $\{+1, -1\}$-Tseitin,*

$$Vars(b = 0) \supseteq \delta(S_b) .$$

Proof. Suppose the two terms of the binomial b are t_1, t_2. Modulo the $\{+1, -1\}$ constraints for the y_e variables, $t_1 + t_2 = 0$ is logically equivalent to $t_1 t_2 = c$, for some field constant c, because these constraints imply that the square of any variable is one, so we may move all variables from t_2 to t_1 or vice versa by multiplying. If both terms are nonzero c will be nonzero, but if one of the terms is zero then c will be zero. However, the case that $c = 0$ is trivial, because already it implies that $b = 0$ contradicts the $\{+1, -1\}$ constraints. So in this case $\mu(b = 0) = 0$ and the claim is vacuous. In fact, this argument shows that $c = \pm 1$. Additionally, we can reduce all terms modulo the squared powers, so that without loss $b = 0$ is of the form $m - c = 0$ for m a monomial with all variables of degree 1 and $c = \pm 1$.

We would like to rephrase the condition Extension constraints $y_e = 1 - 2x_e, \{PARITY_v\}_{v \in S} \models m - c = 0$ so that we can apply Lemma G.4 and be done. To do this, we reverse our "fourier transform" dictionary. As we saw before, products such as m correspond to sums under this transform, so modulo the extension variable constraints, $m = \pm c$ is the same as $\oplus_{e : y_e \in Vars(m)} x_e = c'$ for c' corresponding to c. We then drop the extension constraints since no other formulas remain in the y_e so the logical consequence follows without them, and so we have a system over $\mathbb{F}_2$ as desired. By Lemma G.4, the transformed image of m contains the variables corresponding to the boundary, so by the definition of the transform, $Vars(m) \supseteq \{y_e : e \in \delta(S_b)\}$, as desired. □

The proof of the "extended degree" property is now exactly the same as the proof of the extended width property, Proposition G.7.

Corollary G.19. *If a graph $G = (V, E)$ satisfies extended isoperimetry with parameters W, t_0, and μ' is the strong complexity measure defined from the Binomial complexity measure via*

$$\mu'(C) = \begin{cases} 0 & \mu(C) < t_0 \\ L & \mu(C) > |V|/2 \\ \log_2(\mu(C)/t_0) & \textit{otherwise} \end{cases}$$

then for any k clauses $C_1, \ldots C_k$ with distinct values under μ' between 0 and L,

$$\left| \bigcup Vars(C_i) \right| \geq \Omega(kW) . \tag{G.3}$$

Now we can prove the main result.

Proof. Let π be any proof of $\tau_G[\oplus]$, in size $T = S(\pi)$ and monomial space $S = MSp(\pi)$.

Divide π into epochs recursively to a recursive depth of h, dividing each epoch into m equal subepochs, h, m to be determined later. Now consider the random restriction ρ. Since $\tau_G[\oplus]\restriction_\rho = \tau_G$ (up to replacing variables with their negations), $\pi\restriction_\rho$ is a refutation of τ_G. Let π' be the induced refutation of $\{+1, -1\}$-Tseitin on G, with induced recursive subdivision into epochs. Choose k such that $Lk^{-h} = k$. Since μ' is a strongly bounded complexity measure, Lemma G.3 implies that the critical set of some induced epoch of π' contains at least k binomials of distinct complexity values, with probability one. Thus, by Corollary G.19 this critical set contains $2k$ monomials which collectively contain at least $\Omega((k-2)W)$ variables. By conservativity of the simulations, the critical set of the inducing epoch in $\pi\restriction_\rho$ contains at least $2k$ monomials which collectively contain at least $\Omega((k-2)W)$ variables.

However, for M any small set of monomials, the probability that $M\restriction_\rho$ contains $2k$ monomials of which collectively contain many variables is seen to be small. For any monomial m, the semantically equivalent clause is killed (restricted to a constant) by ρ if and only if m is. Consider now a set of $2k$ monomials. They

all survive ρ if and only if the disjunct of their corresponding clauses does, and this disjunct contains every variable that any one of themc ontains. Therefore by Lemma B.11, the probability that any fixed $2k$-tuple of monomials all survive and have collectively more than X variables is at most exponenitally small in X. By a union bound over all $2k$-tuples of M, the probability that any $2k$ of M's monomials have them have collectively $\Omega((k-2)W)$ variables after the restriction is at most $M^{2k}2^{-\Omega((k-2)W)}$.

Choose the parameter m so that $mS = T/m^{h-1}$, so that all critical sets of epochs have the same size. Choose h so that $h = k$ and $k^h = L$. The probability that any critical set of any epoch contains k complexities after the restriction is thus at most $\left((mS)^2 2^{-\Omega(W)}\right)^k$, and there are at most $m^h = m^k$ epochs. By a union bound, the probability that any critical set of any epoch contains k complexities is at most $\left(m^3 S^2 2^{-\Omega(W)}\right)^k$. Since we know this must happen with certainty, this probability is at least one. We conclude $m \geq \left(2^{\Omega(W)}/S\right)$, or $T \geq \left(2^{\Omega(W)}/S\right)^{\Omega(k)}$. Here k is such that $k^k = L$, so in terms of our L our exponent exceeds $\log L/\log\log L$, as desired. □

References

[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 appeared in *STOC '00*.

[ACL+12] Eric Allender, Shiteng Chen, Tiancheng Lou, Periklis Papakonstantinou, and Bangsheng Tang. Width-parameterized SAT: Time-space tradeoffs. Technical Report TR12-027, Electronic Colloquium on Computational Complexity (ECCC), March 2012.

[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 appeared in *CCC '03*.

[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 appeared in *SAT '09*.

[AR02] Michael Alekhnovich and Alexander A. Razborov. Satisfiability, branch-width and tseitin tautologies. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '02)*, pages 593–603, November 2002.

[AR03] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Nonbinomial 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 appeared in *FOCS '01*.

[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)*, May 2012. To appear.

[BD09] Michael Brickenstein and Alexander Dreyer. PolyBoRi: A framework for Gröbner-basis computations with Boolean polynomials. *Journal of Symbolic Computation*, 44(9):1326–1345, September 2009.

[BDG+09] Michael Brickenstein, Alexander Dreyer, Gert-Martin Greuel, Markus Wedler, and Oliver Wienand. New developments in the theory of Gröbner bases and applications to formal verification. *Journal of Pure and Applied Algebra*, 213(8):1612–1635, August 2009.

[Bea04] Paul Beame. Proof complexity. In Steven Rudich and Avi Wigderson, editors, *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*, pages 199–246. American Mathematical Society, 2004.

[Ben09] Eli Ben-Sasson. Size space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, May 2009. Preliminary version appeared 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 appeared in *CCC '01*.

[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 appeared in *CCC '99*.

[BHvMW09] Armin Biere, Marijn J. H. Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009.

[BI10] Eli Ben-Sasson and Russell Impagliazzo. Random CNF's are hard for the polynomial calculus. *Computational Complexity*, 19:501–519, 2010. Preliminary version appeared in *FOCS '99*.

[BIW04] Eli Ben-Sasson, Russell Impagliazzo, and Avi Wigderson. Near optimal separation of treelike and general resolution. *Combinatorica*, 24(4):585–603, September 2004.

[BJ10] Eli Ben-Sasson and Jan Johannsen. Lower bounds for width-restricted clause learning on small width formulas. 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 16–29. Springer, July 2010.

[BKPS02] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. The efficiency of resolution and Davis-Putnam procedures. *SIAM Journal on Computing*, 31(4):1048–1075, 2002. Preliminary versions of these results appeared in *FOCS '96* and *STOC '98*.

[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. Full-length version available at http://eccc.hpi-web.de/report/2010/125/.

[BP96] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS '96)*, pages 274–282, October 1996.

[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 appeared 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.

[CK02] Peter Clote and Evangelos Kranakis. *Boolean Functions and Computation Models*. Springer, 2002.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC '71)*, pages 151–158, 1971.

[Coo85] Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64(1–3):2–21, 1985.

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

[CS76] Stephen A. Cook and Ravi Sethi. Storage requirements for deterministic polynomial time recognizable languages. *Journal of Computer and System Sciences*, 13(1):25–37, 1976.

[CS80] David A. Carlson and John E. Savage. Graph pebbling with many free pebbles can be difficult. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC '80)*, pages 326–332, 1980.

[CS82] David A. Carlson and John E. Savage. Extreme time-space tradeoffs for graphs with small space requirements. *Information Processing Letters*, 14(5):223–227, 1982.

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

[DLL62] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, July 1962.

[DP60] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.

[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*.

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

[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.

[Her08] Alexander Hertel. *Applications of Games to Propositional Proof Complexity*. PhD thesis, University of Toronto, May 2008. Available at http://www.cs.utoronto.ca/~ahertel/.

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

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

[JMNŽ12] Matti Järvisalo, Arie Matsliah, Jakob Nordström, and Stanislav Živný. Relating proof complexity measures and practical hardness of SAT. Submitted, 2012.

[LT82] Thomas Lengauer and Robert Endre Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *Journal of the ACM*, 29(4):1087–1130, October 1982. Preliminary version appeared in *STOC '79*.

[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.

[NH08] Jakob Nordström and Johan Håstad. Towards an optimal separation of space and length in resolution (Extended abstract). In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pages 701–710, May 2008.

[Nor09a] Jakob Nordström. Narrow proofs may be spacious: Separating space and width in resolution. *SIAM Journal on Computing*, 39(1):59–121, May 2009. Preliminary version appeared in *STOC '06*.

[Nor09b] Jakob Nordström. A simplified way of proving trade-off results for resolution. *Information Processing Letters*, 109(18):1030–1035, August 2009. Preliminary version appeared in ECCC report TR07-114, 2007.

[Nor12a] Jakob Nordström. New wine into old wineskins: A survey of some pebbling classics with supplemental results. Manuscript in preparation. To appear in *Foundations and Trends in Theoretical Computer Science*. Current draft version available at http://www.csc.kth.se/~jakobn/research/, 2012.

[Nor12b] Jakob Nordström. On the relative strength of pebbling and resolution. *ACM Transactions on Computational Logic*, 13(2), 2012. To appear. Preliminary version appeared in *CCC '10*.

[Nor12c] Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 2012. To appear. Available at http://www.csc.kth.se/~jakobn/research/.

[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 appeared in *CP '09*.

[PI00] Pavel Pudlák and Russell Impagliazzo. A lower bound for DLL algorithms for k-SAT (preliminary version). In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '00)*, pages 128–136, January 2000.

[Pip80] Nicholas Pippenger. Pebbling. Technical Report RC8258, IBM Watson Research Center, 1980. Appeared in Proceedings of the 5th IBM Symposium on Mathematical Foundations of Computer Science, Japan.

[Pud97] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, September 1997.

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

[Raz01] 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 2001.

[Rob65] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[SAT] The international SAT Competitions. http://www.satcompetition.org.

[Seg07] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):482–537, December 2007.

[Tse68] Grigori Tseitin. On the complexity of derivation in propositional calculus. In A. O. Silenko, editor, *Structures in Constructive Mathematics and mathematical Logic, Part II*, pages 115–125. Consultants Bureau, New York-London, 1968.

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