Some Trade-off Results for Polynomial Calculus

[Extended Abstract]

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

ABSTRACT

We present size-space trade-offs for the polynomial calculus (PC) and polynomial calculus resolution (PCR) proof systems. These are the first true size-space trade-offs in any algebraic proof system, showing that size and space cannot be simultaneously optimized in these models. We achieve this by extending essentially all known size-space trade-offs for resolution to PC and PCR. As such, our results cover space complexity from constant all the way up to exponential and yield mostly superpolynomial or even exponential size blow-ups. Since the upper bounds in our trade-offs hold for resolution, our work shows that there are formulas for which adding algebraic reasoning on top of resolution does not improve the trade-off properties in any significant way.

As byproducts of our analysis, we also obtain trade-offs between space and degree in PC and PCR exactly matching analogous results for space versus width in resolution, and strengthen the resolution trade-offs in [Beame, Beck, and Impagliazzo '12] to apply also to k-CNF formulas.

Categories and Subject Descriptors: F.2.3[Analysis of Algorithms and Problem Complexity]: Tradeoffs among Complexity Measures; F.1.3[Computation by Abstract Devices]: Complexity Measures and Classes —*Relations among complexity measures*; I.2.3[Artificial Intelligence]: Deduction and Theorem Proving; F.4.1[Mathematical Logic and Formal Languages]: Mathematical Logic —*computational logic*

General Terms: Theory

Keywords: Proof complexity, polynomial calculus, PCR, resolution, trade-offs, size, space, degree, pebble games, pebbling formulas, Tseitin formulas

1. INTRODUCTION

The satisfiability problem is of paramount importance to theoretical computer science. Despite a strong belief in the theory community that the problem is intractable in the worst case, in practice there are many important and successful approaches to solving it. Applied SAT solving has matured significantly in the last 10–15 years, and SAT solvers are now routinely used to solve real-world instances with hundreds of thousands, or even millions, of variables. Today, practitioners often think of SAT as an easy problem to reduce to, rather than a hard problem to reduce from.

Because the SAT algorithms used in practice depend crucially on complex heuristics, essentially the only known way to analyze their worst-case performance is by means of proof complexity. In this approach, the detailed heuristics are abstracted away, and instead the focus is on the proofs which these algorithms generate. Such proofs can be thought of as summarizing the transcripts of the computations, containing only the *reasoning* which took place. Despite this apparently significant loss of information, proof complexity nevertheless has managed to give tight exponential lower bounds on the worst-case running time on approaches for SAT used in practice by lower-bounding proof size. Note that there are many other results in other areas of a similar flavour—rather than directly trying to give lower bounds against, e.g., Turing machines and circuit families, one considers models which contain a canonical algorithm as well as all "nearby" algorithms in some sense. Instead of finding a concrete "bad example" against one algorithm, which might potentially be fixed or avoided, lower bounds in this style show that the whole approach has inherent limitations. For comparison, see, e.g., other work on linear programming hierarchies [20], semidefinite programming hierarchies [39], and algorithmic paradigms [2].

One important recent direction in proof complexity concerns size-space trade-offs. This research is partly driven by concerns about time and memory usage of SAT solvers—in practice, space consumption can be almost as much of a bottleneck as running time—but is also motivated by the fundamental importance of time and space complexity in computation. Time-space trade-offs have historically been one of the most productive directions in computational complexity, and there have been many results in both Boolean and algebraic settings. Typically, the strongest such results show that in a variety of models, the product of time and space in any computation of some function is at least *nearly quadratic* in the input length, giving lower bounds against sublinear space. However, in many applications one has much more than just linear space available, and it is natural to ask whether one can show that there are problems that are solvable in polynomial time but for which any polynomial-time computation must require a large polynomial amount of space. The trade-off results presented in the current paper are the first in an algebraic setting which obtain superpolynomial time blow-up even in the superlinear space regime and also exponential blow-up for sublinear (but polynomial) space, and which still take place in a model which captures practical algorithms.

Our focus in this paper is on the proof systems *polynomial calculus* and *resolution*. Resolution is arguably the most well-studied proof system in proof complexity, and is directly connected with modern SAT solvers based on DPLL [25, 24] with clause learning, also known as *conflict-driven clause-learning (CDCL)* solvers [6, 31]. These algorithms use heuristic-driven backtracking search combined with a dynamic programming technique, and so far have clearly been the most successful approach to solving SAT in practice. Polynomial calculus instead takes an algebraic view of SAT. In this proof system, the disjunctive clauses of a CNF formula are translated into polynomials, and computations in the ideal generated by these polynomials show whether they have a common root or not, corresponding to a satisfying assignment for the formula. It was shown in [22] that such an algebraic approach might be significantly better than resolution-based approaches on some instances, and would never be much worse, so it might ultimately lead to a more "well-rounded" SAT solver. Phrased in the language of proof complexity, the extra expressive power of polynomials can lead to significantly more efficient proofs in terms of size, and perhaps also space. It was suggested in [22] that if our understanding of suitable heuristics could be improved, these algebraic techniques could become competitive with resolution and provide a way around some of the bottlenecks encountered for CDCL solvers.

Intriguingly, however, despite significant progress on algebraic SAT solvers such as PolyBori[18], the gains of the polynomial approach anticipated by [22] have largely failed to materialize. Our research sheds light on one aspect of this, in that it investigates deeper the question of how much the expressive power of polynomials could reasonably be expected to translate into computational efficiency. We show that essentially all time-space trade-offs known for resolution also extend to polynomial calculus, and even to the stronger proof system *polynomial calculus resolution (PCR)* that unifies polynomial calculus and resolution, thus casting doubt on hopes of a generic improvement obtained by using polynomials. Based on what we know now, there seems not to exist any generic transformation of PCR proofs which improves time, improves space, or trades time and space in a way which outperforms what is possible in resolution.

We remark that the issue of time-space trade-offs for SAT is also connected to recent work of the third author on width-parameterized SAT [4], and our improved lower bounds help to strengthen the support contributed by [8] to their thesis.

For more information about proof complexity in general two good references are [7, 40], while the upcoming survey [33] by the second author focuses specifically on time-space trade-offs. A recent, comprehensive reference on SAT solving is [15].

1.1 Previous Work

The resolution proof system appeared in [16] and began to be investigated in connection with automated theorem proving in the 1960s [24, 25, 38]. Despite the apparent simplicity of this proof system, the first superpolynomial lower bounds on proof size were obtained only in 1985 [28] after decades of study. Truly exponential size lower bounds were later proven in [21, 42]. The repertoire of size lower bound techniques remains fairly limited, however, including random restrictions [9, 28], the size-width method [14], and the pseudowidth technique first employed in [35] and further developed in [37].

Polynomial calculus was defined in [22]. In a technical break-through, [36] obtained degree lower bounds. This result was simplified by [30], who also showed that degree lower bounds imply proof size lower bounds in polynomial calculus.

The study of space in resolution was initiated in [26] and was later extended to a more general setting including other proof systems in [1]. Intuitively, the (clause) space of a resolution proof is the maximal number of clauses one needs to keep in memory while verifying the proof. Perhaps somewhat surprisingly, it turns out that linear space is enough to refute any unsatisfiable CNF formula, and a sequence of papers [1, 11, 26] have proven matching lower bounds.

Regarding trade-offs between size and space, some results in restricted settings were obtained in [10, 32] and strong trade-offs for full, unrestricted resolution were reported in the paper [13] involving the second author. These trade-offs only apply for space smaller than the linear worst-case upper bound, however. The recent work [8] by the first author with co-authors presented trade-off results that extend even to superlinear space.

Turning to polynomial calculus and PCR, the space measure (measuring the number of monomials, which is the natural generalization of clause space in resolution) has been quite poorly understood until very recently. While nontrivial space lower bounds were established already in [1], these bounds crucially work only for formulas of unbounded width, and it was only in [27] that space lower bounds for k-CNF formulas were shown. In a very recent paper [17] building on and developing the techniques in [1, 27], optimal linear lower bounds on PCR space were finally obtained, but many intriguing questions about this measure remain open.

As to trade-offs between size and space, we are not aware of any such results for PC or PCR except the recent paper [29] involving the second author. An important distinction here, however, is that in order to speak about a "true" trade-off we want to find formulas which have proofs in small size and also in small space, but for which any proof optimizing one of the measures provably has to pay a stiff penalty with respect to the other measure. While [29] exhibits formulas for which any proofs in small space must have very large size, no such small-space proofs are known to exist. (In fact, it would seem more likely that there are no small-space proofs for these formulas and that the small-size proofs are also optimal with respect to space—this is known to be the case in resolution for very similar formulas.)

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 known to be analogous to that between length and width in resolution, it is open whether monomial space and degree behave with respect to each other as clause space and width do in resolution.

1.2 Our Results

In this paper, we extend the trade-offs in [8, 10, 13], i.e., essentially all known trade-offs for resolution, 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 Sections 2 and 3 for definitions of terminology and notation used below.)

THEOREM 1. *There is a family of explicitly constructible* 3*-CNF formulas* F_n *of size* $\Theta(n)$ *that can be refuted in polynomial calculus in degree* $Deg_{\mathcal{PC}}(F_n \vdash \perp) = \mathrm{O}(1)$ *and also in monomial space* $Sp_{\mathcal{PC}}(F_n \vdash \perp) = \mathrm{O}(1)$*, but such that for any PCR refutation* $\pi_n : F_n \vdash \perp$ *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 [10] and doing a careful analysis of the proofs in [13], which yields a very useful generalization of the techniques there.

Our first set of time-space trade-off results follow by applying the same generalization of [13] to other pebbling formulas. Combining this with random restrictions, we obtain trade-offs where the upper bounds hold 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 [13], 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 obtained are still fairly dramatic, though, 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.

As in [13], we get a whole collection of trade-offs, and we only give two concrete examples here. The first example is that for arbitrarily small but growing space complexity, there can be superpolynomial size-space trade-offs for PC and PCR.

THEOREM 2. *Let* $g(n) = \omega(1)$ *be any arbitrarily slowly growing function*[1] *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* $\mathrm{O}(g(n))$*.*
- *There are PC refutations* π_n *of* F_n *in simultaneous size* $\mathrm{O}(n)$ *and total space* $\mathrm{O}\left(\left(n/g(n)^2\right)^{\frac{1}{3}}\right)$*.*
- *Any PCR refutation of* F_n *in* $\mathrm{O}\left(\left(n/(g(n)^3 \log n)\right)^{\frac{1}{3}-\varepsilon}\right)$ *monomial space must have superpolynomial size.*

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

As a second example, we state a trade-off where the proof size blows up exponentially when space is optimized.

THEOREM 3. *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. F_n *is refutable in PC in total space* $\mathrm{O}\left(n^{1/11}\right)$*.*
2. *There are PC refutations* π_n *of* F_n *in simultaneous size* $\mathrm{O}(n)$ *and total space* $\mathrm{O}\left(n^{3/11}\right)$*.*
3. *Any PCR refutation of* F_n *in monomial space at most* $n^{2/11}/(10 \log n)$ *must have size at least* $\left(n^{1/11}\right)!$ *.*

As in [13], 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, however. For our second set of time-space trade-off results, we instead study so-called Tseitin formulas and lift the trade-offs in [8] 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 if the space is reduced below a polynomial factor of the size of the smallest known proofs, the size must grow as a superconstant power of the optimal size. Besides strengthening this result to PCR, we modify the construction and simplify the technical analysis significantly, which allows us to obtain our trade-offs for 8-CNF formulas, and not just for CNF formulas of unbounded width as in [8].

THEOREM 4. *Let* $\mathbb{F}$ *be a field of odd characteristic. There is an explicitly constructible family of* 8*-CNF formulas* $\{F_{n,w}\}$*, with* $1 \leq w \leq n^{1/4}$*, which are of size* $\Theta(n)$ *and have the following properties:*

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

In fact, the parameter w in $F_{n,w}$ is the *tree-width* of the formula, and this is the reason for the connection with [4] discussed above. In this paper, it was shown that the resolution upper bounds in Theorem 4 can in fact be obtained by a tree-width based algorithm with little overhead, and furthermore that a smooth trade-off upper bound exists between the two ranges. It was conjectured in [4] that this algorithm cannot be improved, which if true would have significant computational complexity consequences.

The lower bounds in Theorem 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 the algorithm in [4]. 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] Technically speaking, we also need $g(n) = \mathrm{O}\left(n^{1/7}\right)$ here, but this restriction is inconsequential since for faster-growing functions we obtain even stronger trade-offs by other means.

1.3 Organization of This Paper

We briefly review preliminaries in Section 2. In Section 3, we give a more detailed overview of our results and describe the main technical ingredients in the proofs. Section 4 contains concluding remarks. Due to space constraints, most of the low-level technical details have had to be omitted, but the formal proofs can be found in the upcoming full-length version.

2. PRELIMINARIES

We consider Boolean formulas over a set of variables $X = \{x_1, \ldots, x_n\}$. A *literal* is a variable x or its negation $\overline{x}$. Sometimes the notation x^1 and x^0 will be handy for unnegated and negated literals, respectively, where x^b is true if $x = b$. A *clause* is a disjunction of literals (without loss of generality over distinct variables), and a *CNF formula* is a conjunction of clauses. We think of clauses as being specified by their sets of literals, and CNFs as specified by their sets of clauses. We write $\mathit{Vars}(C)$ to denote the set of variables appearing in a clause C. The *width* $W(C)$ of a clause C is $|\mathit{Vars}(C)|$ and the width of a formula (or sequence of clauses) F is $\max_{C \in F}\{W(C)\}$. The *size* of a CNF formula F is the total number of literal occurrences, i.e., $\sum_{C \in F} W(C)$.

The *resolution* proof system operates with clauses and has one rule of inference, the resolution rule

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

A *resolution refutation* of a CNF formula is a sequence of clauses ending in the unsatisfiable empty clause $\bot$, where each clause is either from the formula (an *axiom*) or follows from two previous clauses by an application of the resolution rule. The term *resolution derivation* is used more generally to refer to any such sequence of clauses that does not necessarily end with $\bot$. Every resolution derivation naturally corresponds to a directed acyclic graph (DAG), in which every clause derived via the resolution rule has a directed edge to a derived clause from each of its antecedents. Note that the same proof DAG can represent many different resolution proofs (depending on the topological sort of the DAG).

In *polynomial calculus (PC)*, clauses are interpreted as multilinear polynomials over some field $\mathbb{F}$. This is done by identifying truth values with the field elements $\{0, 1\}$, adding for every variable the *Boolean axiom* $x^2 - x$, and translating clauses into polynomials in the natural way. The resulting set of polynomials have a common root—i.e., a satisfying assignment—if and only if the ideal they generate contains 1. A PC refutation is a derivation of 1 using the derivation rules

$$\frac{p}{x \cdot p} \quad \text{and} \quad \frac{p \qquad q}{\alpha p + \beta q} \tag{2}$$

for $\alpha, \beta \in \mathbb{F}$.

We will mostly focus on a common extension of polynomial calculus and resolution called *polynomial calculus resolution (PCR)* [1]. In PCR, we have two distinct formal variables for positive and negative literals over a variable, together with the *complementarity axiom* $x + \overline{x} - 1$ enforcing that these two variables take complementary truth values. This permits the direct simulation of resolution with one monomial per clause.

The *size* of a PC or PCR refutation is measured as the total number of monomials in the refutation (counted with repetitions), whereas *length* is the total number of derived polynomials. Every unsatisfiable k-CNF formula trivially has a refutation of length $\mathrm{O}(n)$, but in general the size of this refutation is exponential.

To measure proof space, we can think of proofs as being presented on a blackboard, where at each step we can write down an axiom, apply an inference rule (to some lines currently written on the blackboard), or erase a line from the blackboard. The space of a proof is then for resolution the maximal number of clauses on the board simultaneously at any time during the proof, and for PC/PCR the maximal number of monomials (counted with repetitions). When studying size-space trade-offs, the size is measured for the same presentation of the proof, where clauses/polynomials appearing multiple times are counted with repetitions.

For any standard definitions, terminology or notation omitted above, we refer to the full-length version of this paper or to [33], which we follow except possibly in minor details.

3. OUTLINE OF RESULTS AND PROOFS

Generally speaking, time-space trade-off results are usually established by some variation of the following plan:

1. Formalize a notion of *work* or *progress* specific to the model and the problem.
2. Divide the time period of a hypothetical computation into a large number of equal-sized *epochs*.
3. Prove the following claims:
   (a) If the epochs are small, then no single epoch makes very much progress.
   (b) If the space is small, then not much progress can be carried over from one epoch to the next.
   (c) To solve the problem, the computation needs to make substantial progress summed over all epochs.
4. Conclude that if the computation is too short and uses too little space, then this leads to a contradiction.

This approach has been implemented in a wide variety of models, including graph pebbling, straight line programs, branching programs, et cetera. To obtain quantitatively strong trade-offs, i.e., trade-offs exhibiting superpolynomial blow-up, in addition it can be necessary to subdivide into epochs *recursively.* Frequently, this kind of refined strategy can only be carried out directly in more limited models. One contribution of our work is that we manage to realize such a strategy in a model which is significantly more general than what has previously been possible. To achieve this, we make careful use of restriction and reduction arguments.

In our first set of trade-offs, which extends the results of [13], we combine random restrictions with a space-faithful projection technique, showing that if there existed PCR refutations which were very efficient with respect to time and space on a certain kind of pebbling formulas, then there would be pebbling strategies for the underlying graphs which would be very efficient as well. Thus we are able to lift graph pebbling lower bounds to PCR.

In fact, our result is more general in that we obtain a kind of generic "hardness amplification" result for CNF formulas. We show that if a formula has a mild form of trade-off in resolution, then by making appropriate syntactic substitutions we obtain another formula which has strong trade-off properties in the stronger proof system PCR. Pebbling then comes into the picture simply because pebbling formulas have exactly the form of weak trade-offs in resolution that we need.

The main technical problem which we overcome is how to reduce PCR refutations of the substituted formulas to resolution refutations of the original formulas in a way that preserves space. In resolution, it is possible to construct space-preserving reductions without using restrictions. Unfortunately, these reductions provably fail for stronger proof systems such as cutting planes and PCR (and even PC), but it turns out that by using random restrictions we can salvage enough of this approach to get strong trade-offs for PCR.

In our second set of trade-offs, which extends the results of [8], we make two contributions. Firstly, we simplify and strengthen the main result in [8] by studying a slightly different formula family with appropriately chosen random restrictions. As a result of this, we can prove trade-offs for k-CNF formulas rather than formulas of asymptotically growing width. Secondly, and more substantially, we manage to implement the overall strategy of [8] in the context of PCR.

This part of our work is different from the generalization of [13] in that it does not obtain a trade-off by reducing to another model of computation—instead, we carry out the plan outlined above directly in the proof system. In [8], this is achieved by using a *semantic measure* of complexity of clauses as a progress measure. One of the crucial technical steps is to prove an inequality showing that not only are clauses representing different progress levels within a certain range all wide, but that they are also "pairwise wide" in that for any pair of such clauses each clause contains many variables not occurring in the other clause. In the context of polynomial calculus, we would need to prove an analogous result using degree instead of width, but sadly such a claim simply is not true.

We circumvent this obstacle by examining the binomial technique of [19] for degree lower bounds in polynomial calculus. This technique is based on the observation that PC refutations of *binomial* systems—i.e., where each initial polynomial is a sum of two monomials—have a special form. Binomial systems are never hard with respect to size or space, but can be hard with respect to degree. The paper [19] obtains degree lower bounds by constructing an explicit pseudoideal, and also gives low-degree reductions from other nonbinomial systems to binomial systems. In this way, it is possible to get degree lower bounds for non-binomial systems, which can in turn be used to obtain size lower bounds.

For our purposes, however, we need much more than just degree lower bounds. We therefore refine the technique of [19] by combining the ideas behind the low-degree reduction and the pseudoideal construction. For any PCR refutation of a Tseitin contradiction, we construct a simulation of it by a restricted form of PC which refutes a "Fourier transformed" version of the formula. This simulation does *not* preserve size or space, but it allows us to obtain a suitable measure of progress for size-space trade-off results. This is because in this restricted setting, the semantic measure is much better behaved, and thanks to this we can prove an analogue of the lemma in [8] discussed above. However, due to the change of variables which occurs it is not true that the simulation commutes with restriction; it is not possible to, e.g., kill the monomials of the "shadow proof" obtained from the simulation with restrictions and argue that the resulting proof simulates the restriction of the original proof. Instead, we use restrictions to eliminate monomials in the original proof, and use key properties of the simulation to show that they cannot reappear in the shadow proof, thus limiting the progress which can be made during its epochs. We can then carry out the progress measurement argument in the shadow proof to obtain a contradiction, given that the original proof was too short and used too little space.

Thus, by carrying out different parts of the argument of [8] in different contexts and mediating between them with this simulation, we are able to establish quantitatively equivalent lower bounds in full PCR. The only other techniques known for degree lower bounds are from [3, 36]; as far as we are aware none of these techniques yield simulations, nor can they be used to obtain time-space trade-offs in the manner described here.

In the rest of this section, we define the formulas under study and elaborate on the proof techniques used.

3.1 Substitution Formulas

Let F be a CNF formula over variables $x, y, z, \ldots$ and let $f : \{0,1\}^d \to \{0,1\}$ be a Boolean function. Then 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) and then expand to conjunctive normal form. We will write $F[f]$ to denote the resulting *substitution formula*. For example, for the disjunctive clause $C = x \vee \overline{y}$ we have

$$\begin{aligned} C[\oplus_2] = {} & (x_1 \vee x_2 \vee y_1 \vee \overline{y}_2) \wedge (x_1 \vee x_2 \vee \overline{y}_1 \vee y_2) \\ & \wedge (\overline{x}_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2) \end{aligned} \tag{3}$$

(where $\oplus_2$ denotes binary exclusive or).

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 preserve resolution and PCR refutations, and so for any refutation $\pi : F[\oplus_2] \vdash \perp$ we have that $\pi\restriction_\rho$ is a refutation of F (modulo sign flips). 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.

3.2 Pebbling Contradictions

Pebbling is a tool for studying time-space relationships by means of a game played on DAGs. Pebble games were originally devised for studying programming languages and compiler construction, but found a broad range of applications in computational complexity during the 70s and 80s, which has expanded further during the last decade to cover also proof complexity. The way pebbling results have been used in proof complexity has mainly been by studying so-called *pebbling contradictions* as defined next.

Definition 5 ([14]). *Let G be a DAG with source vertices S and a unique sink vertex z. Identify every vertex $v \in V(G)$ with a variable v. The* pebbling contradiction *Peb_G over 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*).*

If G has n vertices and indegree ℓ, then Peb_G is an unsatisfiable $(1+\ell)$-CNF formula with $n+1$ clauses over n variables.

3.3 Trade-offs Based on Pebbling

A paradigm that has turned out to be fruitful 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 3.1, and then use this new formula family to prove the desired result. In particular, the time-space trade-offs in [13] are obtained in this way. The techniques in [13] were developed specifically for resolution and the more general k-DNF resolution proof system, but a careful analysis of the proofs reveals that most of the approach can be carried over 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 [13]. 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 [13].

The idea is as follows: Start with a CNF formula F which has a (weak) trade-off in resolution between length and variable space (i.e., the number of variables that any refutation must mention simultaneously at some point). Consider some proof system $\mathcal{P}$ and study the substitution formula $F[f]$, where f is chosen 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 π_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 look at the $\mathcal{P}$-configurations (i.e., snapshots of the blackboard in $\mathcal{P}$-refutations), define *projections* of these $\mathcal{P}$-configurations to clauses over $\mathit{Vars}(F)$, and then to show that such projections translate $\mathcal{P}$-refutations to resolution refutations. Roughly, our intuition 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 convenient 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 [13], we show that any function satisfying the following properties will make this approach work.

Definition 6. *Let f be a d-ary Boolean function. Let $\mathcal{P}$ be a sequential implicational*[2] *proof system with space measure $Sp(\cdot)$, and let $\mathbb{D}$ be any $\mathcal{P}$-configuration over $\mathit{Vars}\big(F[f]\big)$. Then a function proj_f mapping $\mathcal{P}$-configurations $\mathbb{D}$ to sets of clauses $\mathbb{C}$ over $\mathit{Vars}(F)$ is an* f-projection *if it is:*

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

Nontrivial: *If $\mathbb{D} = \emptyset$, then $\mathit{proj}_f(\mathbb{D}) = \emptyset$.*

Monotone: *If $\mathbb{D}' \vDash \mathbb{D}$ and $C \in \mathit{proj}_f(\mathbb{D})$, then C is in or 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 $a \in \mathit{Lit}(A) \backslash \mathit{Lit}(C)$ 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.*

In order for a projection to be of use, it should also somehow preserve space when going from the proof system $\mathcal{P}$ to resolution. This is captured by the next definition.

Definition 7. *We say that proj_f is* space-faithful of degree K *with respect to $\mathcal{P}$ if there is a degree-K polynomial Q such that $Q(Sp(\mathbb{D})) \geq \big|\mathit{Vars}(\mathit{proj}_f(\mathbb{D}))\big|$ holds for any $\mathcal{P}$-configuration $\mathbb{D}$. If $Q(n) = n$, proj_f is* exactly space-faithful.

We show that 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 $\mathcal{P}$. This means that in order to prove time-space trade-offs for, say, PCR or cutting planes, it would be sufficient to design space-faithful projections 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 an important technical contribution.

Unfortunately, for both PCR and cutting planes it seems very challenging to come up with space-faithful projections with respect to the most interesting space measures in these systems. However, there is a particular 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 [13] 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 8. *Let $\mathcal{P}$ be any sequential implicational proof system 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.*

This simple but powerful lemma turns out to be sufficient to lift the resolution trade-offs between width and clause space in [10] to the PCR trade-offs between degree and monomial space in Theorem 1.

The next step is to combine Lemma 8 with substitution using exclusive or. If π is a PCR refutation of $F[\oplus_2]$, then after hitting π with a restriction ρ as described above we get a PCR refutation of the original formula F that is very 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 [13] for resolution, as stated next.

Theorem 9. *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.*[3] *Then any semantic PCR refutation of $F[\oplus_2]$ in monomial space at most $s/\log_{4/3} T$ must have size larger than T.*

[2] Briefly, we say that $\mathcal{P}$ is *sequential implicational* if a $\mathcal{P}$-refutation π is a *sequence* of lines $\pi = \{L_1, \ldots, L_\tau\}$ where each line is semantically implied by previous lines. Note that, e.g., extended Frege does *not* satisfy this property, since introducing a new extension variable as a shorthand for a formula declares an equivalence that is not the consequence of this formula, but cutting planes, PC and PCR do.

[3] It would have been nice to be able to use bounds on refutation length here rather than bounds on the number of axiom

Proof. Let $\pi : F \vdash \bot$ be a PCR refutation of $F[\oplus]$ in size T and monomial space s'. If we apply a random restriction ρ to $F[\oplus]$ as described above, then $\pi\lceil_\rho$ is a PCR refutation of F. Consider some fixed monomial m in π. It is easy to show that $m\lceil_\rho$ has degree at most K except with probability $(3/4)^K$. Thus, by a union bound we can pick ρ so that $\pi\lceil_\rho$ 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 8, this results in a resolution refutation 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 2 and 3, as well as several other trade-off results, now follow by applying Theorem 9 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, even when this system is made stronger still by allowing semantic derivation steps.

3.4 Tseitin Contradictions

Tseitin contradictions [41] encode the principle that every undirected graph has even total degree.

Definition 10. *Let $G = (V, E)$ be an undirected graph and $\chi : V \to \{0, 1\}$ a function such that $\bigoplus_{v \in V} \chi(v)$ is odd. Identify each edge $e \in E$ with a variable x_e, and for a vertex $v \in V$ and value $b \in \{0, 1\}$ let*

$$PARITY_{v,b} = \bigwedge \left\{ \bigvee_{e \ni v} x_e^{a(e)} \;\middle|\; \bigoplus_e (a(e) \oplus 1) \neq b \right\}$$

be the CNF representation of the constraint $\bigoplus_{e \ni v} x_e = b$. Then the Tseitin contradiction *on (G, χ) is*

$$Ts(G, \chi) = \bigwedge_{v \in V} PARITY_{v, \chi(v)} \quad .$$

Since each edge is counted twice in $Ts(G, \chi)$, the parity constraints cannot all be satisfied if the overall parity of χ is odd. We will frequently suppress the reference to χ above, since when G is connected any two odd-parity functions yield equivalent formulas for all practical purposes.

When the degree of the graph is bounded by d, each local parity constraint for a vertex can be written as a CNF formula with at most 2^{d-1} clauses of width d, and hence $Ts(G)$ has at most $2^{d-1}|V|$ clauses in total.

3.5 Trade-offs for Tseitin Contradictions

A useful tool when proving lower bounds in resolution are *semantic measures* of clause complexity as introduced by Ben-Sasson and Wigderson [14], i.e., measures of the form

$$\mu_{\mathcal{A}}(C) = \min\{|S| : S \subseteq \mathcal{A},\, S \models C\} \ , \qquad (4)$$

where C is a clause and $\mathcal{A}$ is a collection of axioms (or sets of axioms). In the context of Tseitin contradictions $Ts(G)$, the semantic measure of a clause is defined to be the size of a smallest subset of the vertices of G such that the parity constraints over these vertices semantically imply the clause. Tseitin contradictions cannot be refuted without using parity constraint clauses for all vertices, so in the course of the proof information from all vertices must be aggregated. If the graph G has a large isoperimetric number—i.e., if every medium-sized set of vertices have many edges leaving the set—then the formula $Ts(G)$ will be hard to refute.

In [8] it was shown how to use the semantic measure as a progress measure to get not only size lower bounds but also size-space trade-offs for Tseitin contradictions in resolution. This strategy considers multiple ranges of intermediate complexity values for clauses, and requires quite specific and strong isoperimetric properties. The argument works for graphs that not only have a certain *extended isoperimetry* property, but also maintain this property after having a constant fraction of randomly chosen edges removed (corresponding to the formula being hit by a random restriction). This creates significant complications at a fairly low level of the argument, and necessitates the use of rather dense graphs, so that the trade-off can only be shown to hold for CNF formulas of unbounded width. We get a cleaner and simpler proof by instead considering multigraphs and using appropriate restrictions operating on them. This makes the argument more transparent and enables us to prove trade-offs for CNF formulas of *constant width* (which, as noted in, e.g., [1], is the preferred setting when studying space in proof complexity).

As in [8], our construction requires graphs with an extended isoperimetry property, which is formalized as follows.

Definition 11. *Let $G = (V, E)$ be an undirected graph and (W, t_0, r) be associated parameters. Call a vertex set $S \subseteq V$ of size $t_0 \leq |S| \leq |V|/2$* medium-sized. *The* boundary *of S is $\delta(S) = \{(u, v) \in E \mid u \in S,\ v \in V \setminus S\}$, i.e., the set of edges with exactly one endpoint in S.*

We say that G has the extended isoperimetry property *with parameters (W, t_0, r) if any sequence of medium-sized sets of vertices $S_1, \ldots, S_k \subseteq V$ such that $|S_{i+1}| \geq r \cdot |S_i|$ for all i satisfies the inequality $\left|\bigcup_i \delta(S_i)\right| \geq k \cdot W$.*

So-called *grid graphs* (with vertices indexed by integer coordinates (i, j) and edges to adjacent coordinates $(i, j \pm 1)$, $(i \pm 1, j)$) can be shown to have this property.

Lemma 12. *A $w \times \ell$ grid graph, where $4w^2 \leq \ell \leq 2^w$, satisfies the extended isoperimetry property with parameters $(w, 4w^3, 2 + \epsilon)$ for any $\epsilon > 0$ and any large enough w.*

For Tseitin contradictions, vertex set boundaries are related to the semantic complexity measure μ as follows.

Lemma 13 ([14]). *Let S be a minimal vertex set witnessing the complexity $\mu(C)$ of a clause C derived from $Ts(G)$ in resolution. Then $\delta(S) \subseteq Vars(C)$.*

We now formalize the idea that the semantic measure μ can be used as a tool to obtain time-space trade-offs. The following lemma is straightforward to prove by induction.

Lemma 14. *Fix any unsatisfiable CNF formula F with associated semantic measure μ_F and any sequential implicational proof system. Let $\mu^*(\cdot) = \lfloor \log_2 \mu_F(\cdot) \rfloor$ and let K_{lo}, K_{hi}*

downloads. This is clearly *not* possible, however. The reason for this is that the proof refuting $F[\oplus_2]$ 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.

be fixed integers. If a refutation of F is divided into consecutive subderivations, or epochs, *and further subdivided recursively into subepochs to a recursive depth of h, then for any integer k at least one of the following cases apply:*

1. *There exists an epoch at a leaf in the recursive tree which contains formulas with at least $(K_{hi} - K_{lo} + 1) \cdot k^{-h}$ distinct values in $[K_{lo}, K_{hi}]$ under μ^*.*
2. *There exists an epoch such that the formulas in memory during the breakpoints between the epochs in its immediate children contain formulas with at least k distinct values in $[K_{lo}, K_{hi}]$ under μ^*.*

To apply this lemma, consider Tseitin formulas over grid graphs G, do $\oplus_2$-substitution in $Ts(G)$ (which yields the formula $Ts(G'')$ over the multigraph G'' with two copies of each edge in G), and hit any resolution refutation of $Ts(G)[\oplus_2] = Ts(G'')$ with a random restriction as described in Section 3.1. Since G satisifies extended isoperimetry, the clauses in Lemma 14 are collectively wide. For this very reason, however, a random restriction would have been very likely to kill at least one of these clauses. Trading off parameters appropriately, we obtain strong time-space trade-offs.

Unfortunately, this strategy breaks down in polynomial calculus. Naively, one would hope to use the analogy between resolution width and PC degree and just carry out the plan above. However, we cannot obtain degree bounds from the semantic measure because it is not true that medium complexity polynomials are necessarily of high degree, and existing degree lower bound techniques do not seem to yield progress measures of the kind needed for Lemma 14.

One such lower bound technique is to do a linear transformation of the variables. In the context of PCR over a field of odd characteristic, consider rewriting the Tseitin parity constraints so that the variables take values in $\{+1, -1\}$ rather than $\{0, 1\}$. It is not hard to see that the degree needed to refute this "Fourier-transformed" $\{\pm 1\}$-Tseitin formula is the same as for the original $\{0, 1\}$-Tseitin formula, and [19] gave tight upper and lower degree bounds for the former exploiting the fact that its polynomials have a simple binomial form. While using this machinery in our setting can easily establish that any refutation must contain many monomials corresponding to boundaries for vertex sets of many different sizes, by itself this does not suffice for time-space trade-offs. We must also be able to show how these monomials are related, so that we can track progress made by a refutation.

We resolve this issue by translating PCR refutations of Tseitin formulas to refutations of the Fourier-transformed formulas in the *binomial polynomial calculus* subsystem used in [19]. It turns out that in binomial PC a suitable progress measure can be found, so the plan above can be carried out there. Thanks to the following key property of the binomial PC simulation of the original refutation we can then translate back again to PCR.

Lemma 15. *The simulation of PCR on $Ts(G)$ (in variables $\{x_e\}$) by binomial PC on the $\{\pm 1\}$-Tseitin formula (in variables $\{y_e\}$) is* conservative with respect to monomials*:*

- *If for some configuration of the simulated refutation no monomial appears which contains the set of variables $\{x_e \mid e \in E'\}$ for some $E' \subseteq E$, then the corresponding configuration of the simulating refutation does not contain any monomials containing all of $\{y_e : e \in E'\}$.*
- *If for some time period in the simulated refutation no monomial contains the set of variables $\{x_e \mid e \in E'\}$ for some $E' \subseteq E$, then the corresponding time period of the simulating refutation does not contain any monomials containing all of $\{y_e : e \in E'\}$.*

Since the simulation is *not* efficient with respect to size and space, the images of epochs under the simulation have wildly differing sizes in general. However, for the binomials we can recover a degree-analogue of the width lower bound in resolution without losing much, as stated next.

Lemma 16. *Suppose that $G = (V, E)$ has the extended isoperimetry property with parameters (w, t_0, r), and as before let $\mu^*(\cdot) = \lfloor \log_2 \mu(\cdot) \rfloor$. Then for any binomials $b_1, \ldots, b_k$ with distinct complexities between t_0 and $\mu^*(\bot)$ it holds that*

$$\left| \bigcup Vars(b_i) \right| \geq \Omega(k \cdot w) \ . \tag{5}$$

Using the semantic measure in binomial PC together with the division of the refutation into epochs in Lemma 14, we can obtain many monomials of collectively high complexity either within a single epoch or at a collection of breakpoints in the simulating refutation. Then we can apply Lemma 15 to lift these monomials back to the simulated refutation where they are unlikely to survive a restriction.

Proof sketch for Theorem 4. Let G be a grid graph satisfying Lemma 16 and let π be any PCR refutation of $Ts(G)[\oplus_2]$ in size $S(\pi) = T$ and space $Sp(\pi) = S$. Divide π into epochs with each epoch split into m equal subepochs to a recursive depth of h, with m and h to be determined later. Say that the *critical set* of monomials associated to an internal epoch consists of any monomial appearing at the breakpoints between its children, and that for a leaf epoch the critical set contains all monomials in the epoch.

Applying the random restriction ρ in Section 3.1, we get that $\pi\!\restriction_\rho$ is a refutation of $Ts(G)[\oplus]\!\restriction_\rho = Ts(G)$ (up to sign flips). Let π^* be the induced refutation of the $\{\pm 1\}$-Tseitin formula on G with corresponding induced recursive subdivision into epochs. Apply Lemma 14 using thresholds $\lfloor \log t_0 \rfloor$, $\lfloor \log |V(G)|/2 \rfloor$, matching the definition of extended isoperimetry. Let I denote the number of intermediate μ^* values. Choose k such that $I \cdot k^{-h} = k$. Combining the properties of μ^* with Lemma 14 implies that the critical set of some induced epoch of π^* contains at least k binomials of distinct complexity values. Thus, by Lemma 16 this critical set contains $2k$ monomials which collectively contain at least $\Omega(k \cdot w)$ variables. By Lemma 15, this holds for π^* also.

However, the restriction $M\!\restriction_\rho$ of any small set of monomials M is unlikely to have $2k$ monomials which collectively contain many variables. The probability that any fixed $2k$-tuple of monomials in the x-variables all survive and have collective width W is at most $\exp(-\Omega(W))$. By a union bound over all $2k$-tuples of M, the probability that any $2k$ of the monomials in M have collectively $\Omega(k \cdot w)$ variables after the restriction is at most $|M|^{2k} \exp(-\Omega(k \cdot w))$.

Choose the parameter m so that $m^h = (T/S)$ and let $K = mS = Tm^{-h+1}$. For this choice of m the sizes of all critical sets of monomials are bounded by K. The probability for any critical set of any induced epoch to contain k distinct complexities with respect to μ^* after the restriction is at most $K^{2k} \exp(-k \cdot w)$, and there are at most m^h epochs. Set $h = k$. By a union bound, the probability that any epoch contains k complexities is at most $(mK^2 \exp(-\Omega(w)))^k$. On the other hand, by Lemmas 14 and 15, this probability is 1. We conclude that $T \geq (\exp(\Omega(w))/S)^{\Omega(h)}$. Since h and k may be set as large as $\log I/\log\log I$, and I may be set as large as $\Omega(\log|V[G]|)$, ultimately gives $T \geq (\exp(\Omega(w))/S)^{\Omega(\log\log n/\log\log\log n)}$. □

We conclude this section by commenting on a novel aspect of this result. While we avoided explicitly constructing a complexity measure for PCR, we obtain an implicit measure lifted from the binomial PC simulation. A little reflection reveals that this progress measure is in fact non-local—the complexity of a polynomial depends on the derivation used to obtain it—in contrast to most measures we have seen in the literature. Perhaps this new technique for constructing progress measures could be useful in other contexts as well.

4. 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, we show that in the worst case it is impossible to do any meaningful simultaneous optimization of size and space. Polynomial calculus and PCR are still not very well understood, however, and there remain several interesting open problems.

One such problem, which has seen exciting developments lately, is to prove lower bounds on space in PCR. It is only very recently that [17, 27] managed to obtain lower bounds for k-CNF formulas, but these bounds all require $k \geq 4$. Intriguingly, there are still no superconstant lower bounds for any 3-CNF formula. Also, for several well-known formula families the space complexity remains open.

Another question is how far the analogies go between size, (monomial) space, and degree in PC/PCR on the one hand and length, (clause) space, and width in resolution on the other. In resolution, we know that clause space is an upper bound on width [5], that small width does not say anything about space complexity [12], and that there can be very strong trade-offs between these two measures [10]. In this paper, we have shown that exactly the same kind of trade-off holds between degree and monomial space in PC and PCR. However, we still 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 based on 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 Section 3.3 together with the right kind of substitutions in the formulas to get tight trade-offs. It would be very interesting 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 also yield new lower bounds on space similar to what is currently known for resolution.

Looking beyond polynomial calculus, another proof system that would be very interesting to understand is cutting planes. Here open problems abound. Perhaps most obviously, it would be desirable to prove size lower bounds by some other technique than the interpolation used in [34] for, say, Tseitin contradictions 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 [29] 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 for cutting planes. If true, this would mean that the so-called black-white pebble game in [23] could be used to obtain strong trade-offs not only for resolution, k-DNF resolution and PC/PCR, but also for cutting planes.

Finally, it is known that PCR and cutting planes are both strictly stronger than resolution with respect to proof size, and it would seem natural to expect that they should both be stronger than resolution with respect to space as well. As far as we are aware, though, this is open. 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 Ron-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 grants 621-2010-4797 and 621-2012-5645 from the Swedish Research Council 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 by National Basic Research Program of China grants 2011CBA00300, 2011CBA00301, NSFC grants no. 61033001, 61073174, 61250110577, and he also acknowledges support from the Danish National Research Foundation and the National Science Foundation of China (under the grant 61061130540) for the Sino-Danish Center for the Theory of Interactive Computation, within which part of this work was performed.

5. REFERENCES

[1] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211, 2002.

[2] M. Alekhnovich, A. Borodin, J. Buresh-Oppenheim, R. Impagliazzo, A. Magen, and T. Pitassi. Toward a model for backtracking and dynamic programming. *Comput. Compl.*, 20(4):1–62, 2012.

[3] M. Alekhnovich and A. A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *Proc. Steklov Institute of Mathematics*, 242:18–35, 2003.

[4] E. Allender, S. Chen, T. Lou, P. Papakonstantinou, and B. Tang. Width-parameterized SAT: Time-space tradeoffs. TR12-027, ECCC, 2012.

[5] A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. *J. Comput. System Sci.*, 74(3):323–334, 2008.

[6] R. J. Bayardo Jr. and R. Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proc. 14th National Conference on Artificial Intelligence*, pp. 203–208, 1997.

[7] P. Beame. Proof complexity. *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*, pp. 199–246, 2004.

[8] P. Beame, C. Beck, and R. Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proc. 44th ACM Symposium on Theory of Computing*, pp. 213–232, 2012.

[9] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Proc. 37th IEEE Symposium on Foundations of Computer Science*, pp. 274–282, 1996.

[10] E. Ben-Sasson. Size space tradeoffs for resolution. *SIAM J. Comput.*, 38(6):2511–2525, 2009.

[11] E. Ben-Sasson and N. Galesi. Space complexity of random formulae in resolution. *Rand. Struct. Alg.*, 23(1):92–109, 2003.

[12] E. Ben-Sasson and J. Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proc. 49th IEEE Symposium on Foundations of Computer Science*, pp. 709–718, 2008.

[13] E. Ben-Sasson and J. Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proc. 2nd Symposium on Innovations in Computer Science*, pp. 401–416, 2011.

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

[15] A. Biere, M. J. H. Heule, H. van Maaren, and T. Walsh, editors. *Handbook of Satisfiability*, vol. 185 of *FAIA*. IOS Press, 2009.

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

[17] I. Bonacina and N. Galesi. Pseudo-partitions, transversality and locality: A combinatorial characterization for the space measure in algebraic proof systems. In *Proc. 4th Conference on Innovations in Theoretical Computer Science*, pp. 455–472, 2013.

[18] M. Brickenstein and A. Dreyer. PolyBoRi: A framework for Gröbner-basis computations with Boolean polynomials. *J. Symb. Comput.*, 44(9):1326–1345, 2009.

[19] S. R. Buss, D. Grigoriev, R. Impagliazzo, and T. Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. System Sci.*, 62(2):267–289, 2001.

[20] M. Charikar, K. Makarychev, and Y. Makarychev. Integrality gaps for sherali-adams relaxations. In *Proc. 41st ACM Symposium on Theory of Computing*, pp. 283–292, 2009.

[21] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988.

[22] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proc. 28th ACM Symposium on Theory of Computing*, pp. 174–183, 1996.

[23] S. A. Cook and R. Sethi. Storage requirements for deterministic polynomial time recognizable languages. *J. Comput. System Sci.*, 13(1):25–37, 1976.

[24] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Comm. ACM*, 5(7):394–397, 1962.

[25] M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[26] J. L. Esteban and J. Torán. Space bounds for resolution. *Info. Comp.*, 171(1):84–97, 2001.

[27] Y. Filmus, M. Lauria, J. Nordström, N. Thapen, and N. Ron-Zewi. Space complexity in polynomial calculus. In *Proc. 27th IEEE Conference on Computational Complexity*, pp. 334–344, 2012.

[28] A. Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39(2-3):297–308, 1985.

[29] T. Huynh and J. Nordström. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Proc. 44th ACM Symposium on Theory of Computing*, pp. 233–248, 2012.

[30] R. Impagliazzo, P. Pudlák, and J. Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Comput. Compl.*, 8(2):127–144, 1999.

[31] J. P. Marques-Silva and K. A. Sakallah. GRASP—a new search algorithm for satisfiability. In *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp. 220–227, 1996.

[32] J. Nordström. A simplified way of proving trade-off results for resolution. *Info. Proc. Lett.*, 109(18):1030–1035, 2009.

[33] J. Nordström. Pebble games, proof complexity and time-space trade-offs. *Log. Meth. Comput. Sci.*, to appear, 2013.

[34] P. Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Logic*, 62(3):981–998, 1997.

[35] R. Raz. Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004.

[36] A. A. Razborov. Lower bounds for the polynomial calculus. *Comput. Compl.*, 7(4):291–324, 1998.

[37] A. A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theor. Comput. Sci.*, 1(303):233–243, 2003.

[38] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.

[39] G. Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proc. 49th IEEE Symposium on Foundations of Computer Science*, pp. 593–602, 2008.

[40] N. Segerlind. The complexity of propositional proofs. *Bull. Symb. Logic*, 13(4):482–537, 2007.

[41] G. Tseitin. On the complexity of derivation in propositional calculus. In *Structures in Constructive Mathematics and mathematical Logic, Part II*, pp. 115–125, 1968.

[42] A. Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987.