From Small Space to Small Width in Resolution

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

1 **Simons Institute for the Theory of Computing, University of California, Berkeley, USA**

2 **School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden**

Abstract

In 2003, Atserias and Dalmau resolved a major open question about the resolution proof system by establishing that the space complexity of formulas is always an upper bound on the width needed to refute them. Their proof is beautiful but somewhat mysterious in that it relies heavily on tools from finite model theory. We give an alternative, completely elementary, proof that works by simple syntactic manipulations of resolution refutations. As a by-product, we develop a "black-box" technique for proving space lower bounds via a "static" complexity measure that works against any resolution refutation—previous techniques have been inherently adaptive. We conclude by showing that the related question for polynomial calculus (i.e., whether space is an upper bound on degree) seems unlikely to be resolvable by similar methods.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes – Relations among complexity measures, F.4.1 Mathematical Logic – Computational logic, F.2.2 Nonnumerical Algorithms and Problems – Complexity of proof procedures

Keywords and phrases proof complexity, resolution, width, space, polynomial calculus, PCR

Digital Object Identifier 10.4230/LIPIcs.STACS.2014.300

1 Introduction

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

Resolution is also intimately connected to SAT solving, in that it lies at the foundation of state-of-the-art SAT solvers using so-called conflict-driven clause learning (CDCL). This connection has motivated the study of *proof space* as a second interesting complexity measure for resolution. The space usage at some step t in a proof is measured as

the number of clauses occurring before C_t that will be used to derive clauses after C_t, and the space of a proof is obtained by taking the maximum over all steps t.

For both of these complexity measures, it turns out that a key role is played by the auxiliary measure of *width*, i.e., the size of a largest clause in the proof. In a celebrated result, Ben-Sasson and Wigderson [10] showed that there are short resolution refutations of a formula if and only if there are also (reasonably) narrow ones, and almost all known lower bounds on resolution length can be (re)derived using this connection.

31st Symposium on Theoretical Aspects of Computer Science (STACS'14).
Editors: Ernst W. Mayr and Natacha Portier; pp. 300–311
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

In 2003, Atserias and Dalmau (journal version in [2]) established that width also provides lower bounds on space, resolving a problem that had been open since the space complexity of propositional proofs started being studied in the late 1990s ([1, 13]). This means that for space also, almost all known lower bounds can be rederived by using width lower bounds and appealing to [2]. This is not a two-way connection, however, in that formulas of almost worst-case space complexity may require only constant width as shown in [8].

The starting point of our work is the lower bound on space in terms of width in [2]. This is a very elegant but also magical proof in that it translates the whole problem to Ehrenfeucht–Fraïssé games in finite model theory, and shows that resolution space and width correspond to strategies for two opposite players in such games. Unfortunately, this also means that one obtains essentially no insight into what is happening on the proof complexity side (other than that the bound on space in terms of width is true). It has remained an open problem to give a more explicit, proof complexity theoretic, argument.

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

This new, simple proof also allows us to obtain a new technique for proving space lower bounds. This approach is reminiscent of [10] in that one defines a static "progress measure" on refutations and argues that when a refutation has made substantial progress it must have high complexity with respect to the proof complexity measure under study. Previous lower bounds on space have been inherently adaptive and in that sense less explicit.

One other important motivation for our work was the hope that a simplified proof of the space-width inequality would serve as a stepping stone to resolving the analogous question for the polynomial calculus proof system, where the width of clauses corresponds to the *degree* of polynomials. While we recently showed in [14] the analogue of [8] that there are formulas of worst-case space complexity that require only constant degree, the question of whether degree lower bounds imply space lower bounds remains open. Unfortunately, as discussed towards the end of this paper we show that it appears unlikely that this question can be resolved by methods similar to our proof of the corresponding inequality for resolution.

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

2 Preliminaries

Let us start by a brief review of the preliminaries. The following material is standard and can be found, e.g., in the survey [18].

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

A clause C' *subsumes* clause C if every literal from C' also appears in C. A *k-clause (k-term)* is a clause (term) that contains at most k literals. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses, and a *DNF formula* $F = T_1 \vee \cdots \vee T_m$ is a disjunction of terms. A *k-CNF formula (k-DNF formula)* is a CNF formula (DNF formula) consisting of k-clauses (k-terms). In this paper we only consider CNF formulas that do not contain the empty clause. We think of clauses, terms, and CNF formulas as sets: the order of elements is irrelevant and there are no repetitions.

Let us next describe a slight generalization of the resolution proof system by Krajíček [16], who introduced the the family of *r-DNF resolution* proof systems, denoted $\mathcal{R}(r)$, as an intermediate step between resolution and depth-2 Frege systems. An *r-DNF resolution configuration* $\mathbb{C}$ is a set of r-DNF formulas. An *r-DNF resolution refutation* of a CNF formula F is a sequence of configurations $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$ such that $\mathbb{C}_0 = \emptyset$, $\bot \in \mathbb{C}_\tau$, and for $1 \le t \le \tau$ we obtain $\mathbb{C}_t$ from $\mathbb{C}_{t-1}$ by one of the following steps:

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

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

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

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

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

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

Erasure $\mathbb{C}_t = \mathbb{C}_{t-1} \setminus \{C\}$, where C is an r-DNF formula in $\mathbb{C}_{t-1}$.

When setting $r = 1$ we obtain the standard *resolution* proof system. In this case the only nontrivial inference rules are weakening and r-cut, where the former can be eliminated without loss of generality (but is sometimes convenient to have for technical purposes) and the latter simplifies to the *resolution rule* $\frac{C \vee x \quad D \vee \overline{x}}{C \vee D}$. We identify a resolution configuration $\mathbb{C}$ with the CNF formula $\bigwedge_{C \in \mathbb{C}} C$.

The *length* $L(\pi)$ of an r-DNF resolution refutation π is the number of download and inference steps, and the *space* $Sp(\pi)$ is the maximal number of r-DNF formulas in any configuration in π. We define the length $L_{\mathcal{R}(r)}(F \vdash \bot)$ and the space $Sp_{\mathcal{R}(r)}(F \vdash \bot)$ of refuting a formula F in r-DNF resolution by taking the minimum over all refutations F with respect to the relevant measure. We drop the proof system $\mathcal{R}(r)$ from this notation when it is clear from context.

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

3 From Space to Width

In this section we present our new combinatorial proof that width is a lower bound for clause space in resolution. The formal statement of the theorem is as follows (where we recall that in this article all CNF formulas are assumed to be non-trivial in that they do not contain the contradictory empty clause).

▶ **Theorem 1** ([2]). *Let F be a k-CNF formula and let $\pi : F \vdash \bot$ be a resolution refutation in space $Sp(\pi) = s$. Then there is a resolution refutation π' of F in width $W(\pi') \leq s + k - 3$.*

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

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

▶ **Definition 2.** The *negated configuration* $\mathrm{neg}(\mathbb{C})$ of a configuration $\mathbb{C}$ is defined by induction on the number of clauses in $\mathbb{C}$:

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

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

In the proof we will use a different characterization of negated configurations that is easier to work with.

▶ **Proposition 3.** *The negated configuration $\mathrm{neg}(\mathbb{C})$ is the set of all minimal (non-trivial) clauses C such that $\neg C$ implies the configuration $\mathbb{C}$. That is*

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

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

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

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

The following observation, which formalizes the main idea behind the concept of negated configurations, is an immediate consequence of Proposition 3.

▶ **Observation 4.** *An assignment satisfies a clause configuration $\mathbb{C}$ if and only if it falsifies the negated clause configuration* $\mathrm{neg}(\mathbb{C})$*. That is,* $\mathbb{C}$ *is logically equivalent to* $\neg\mathrm{neg}(\mathbb{C})$*.*

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

▶ **Lemma 5.** *If* $\mathbb{C} \vDash \mathbb{C}'$*, then for every clause* $C \in \mathrm{neg}(\mathbb{C})$ *there is a clause* $C' \in \mathrm{neg}(\mathbb{C}')$ *such that* C *is a weakening of* C'*.*

Proof. For any clause C is in $\mathrm{neg}(\mathbb{C})$ it holds by Proposition 3 that $\neg C \vDash \mathbb{C}$. Since $\mathbb{C} \vDash \mathbb{C}'$, this in turns implies that $\neg C \vDash \mathbb{C}'$. Applying Proposition 3 again, we conclude that there exists a clause $C' \subseteq C$ such that $C' \in \mathrm{neg}(\mathbb{C}')$. ◀

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

Proof of Theorem 1. Let $\pi = (\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau)$ be a resolution refutation of F in space s. For every configuration

$\mathbb{C}_t \in \pi$, let $\mathbb{D}_t$ denote the corresponding negated configuration $\mathrm{neg}(\mathbb{C}_t)$. By the discussion preceding Definition 2, it is clear than each clause of $\mathbb{C}_t$ contributes at most one literal to each clause of $\mathbb{D}_t$. Hence, the clauses of $\mathbb{D}_t$ have width at most s. We need to show how to transform the sequence $\pi' = (\mathbb{D}_\tau, \mathbb{D}_{\tau-1}, \ldots, \mathbb{D}_0)$ into a legal resolution refutation of width at most $s + k - 3$.

The initial configuration of the new refutation is $\mathbb{D}_\tau$ itself, which is empty by Definition 2. If $\mathbb{C}_{t+1}$ follows $\mathbb{C}_t$ by inference or erasure, then we can derive any clause of $\mathbb{D}_t$ from a clause of $\mathbb{D}_{t+1}$ by weakening, as proven in Lemma 5. If $\mathbb{C}_{t+1}$ follows $\mathbb{C}_t$ by axiom download, then we can derive $\mathbb{D}_t$ from $\mathbb{D}_{t+1}$ in width at most $s+k-3$, as we show below. The last configuration $\mathbb{D}_0$ includes the empty clause $\perp$ by Definition 2, so the new refutation is complete.

It remains to take care of the case of axiom download. We claim that we can assume without loss of generality that prior to each axiom download step the space of the configuration $\mathbb{C}_t$ is at most $s - 2$. Otherwise, immediately after the axiom download step the proof π needs to erase a clause in order to maintain the space bound s. By reordering the axiom download and clause erasure steps we get a valid refutation of F for which it holds that $Sp(\mathbb{C}_t) \leq s - 2$.

Suppose $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{A\}$ for some axiom $A = a_1 \vee \cdots \vee a_\ell$, with $\ell \leq k$. Consider now some clause C that is in the negated configuration $\mathbb{D}_t$ and that does not belong to $\mathbb{D}_{t+1}$. Again by Definition 2, the clause C has at most one literal per clause in $\mathbb{C}_t$, so $W(C) \leq s - 2$. To derive C from $\mathbb{D}_{t+1}$ we first download axiom A and then show how to derive C from the clauses in $\mathbb{D}_{t+1} \cup \{A\}$.

First, note that all clauses $C_a = C \vee \overline{a}$ are either contained in or are weakenings of clauses in $\mathbb{D}_{t+1}$. This follows easily from Definition 2 as adding an axiom A to the configuration $\mathbb{C}_t$ results in adding negations of literals from A to all clauses $C \in \mathbb{D}_t$. Hence, we can obtain C by the following derivation:

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

$$\vdots$$

$$\dfrac{C \vee a_\ell \qquad\qquad C_{a_\ell} = C \vee \overline{a}_\ell}{C}$$

When C is the empty clause, the width of this derivation is upper-bounded by $W(A) \leq k$. Otherwise, it is upper bounded by $W(C) + W(A) - 1 \leq s + k - 3$. Any resolution refutation has space at least 3 (unless the formula contains the empty clause itself), so the width of π' is upper-bounded by $W(\pi') \leq s + k - 3$. ◀

The proof of Theorem 1 also works for r-DNF resolution, with some loss in parameters. We now define the negated configuration of an r-DNF resolution configuration and sketch a proof that resolution width is a lower bound for r-DNF resolution space.

▶ **Theorem 6.** *Let F be a k-CNF formula and let $\pi : F \vdash \perp$ be an r-DNF resolution refutation of F in space $Sp(\pi) \leq s$. Then there exists a resolution refutation π' of F in width at most $W(\pi') \leq (s-2)r + k - 1$.*

Proof sketch. We define the negated configuration $\mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C})$ of a $\mathcal{R}(r)$ configuration to be

- $\mathrm{neg}_{\mathcal{R}(r)}(\emptyset) = \{\perp\}$,
- $\mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C} \cup \{C\}) = \{D \vee \overline{T} \mid D \in \mathrm{neg}_{\mathcal{R}(r)}(\mathbb{C}) \text{ and } T \in C\}$,

with trivial and subsumed clauses removed. It is easy to see that an s space r-DNF configuration gets transformed into a resolution configuration of width sr. We can prove an analogue of Proposition 3 for this definition of the negated configuration and, hence, the analogue of Lemma 5 easily follows. The case of axiom download is the same as in the proof of Theorem 1 as axioms are clauses. Hence, running the negated refutation backwards we get a resolution refutation of F in width $(s-2)r + k - 1$. ◀

4 A Static Technique for Proving Space Lower Bounds

Looking at the proof complexity literature, the techniques used to prove lower bounds for resolution length and width (e.g., [10, 11, 15, 19]) are essentially different from ones used to prove resolution space lower bounds (e.g., [1, 7, 13], in that the former are *static* or *oblivious* while the latter are *dynamic*.

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

In contrast, most lower bound proofs for resolution space use an *adversary argument.* Assuming that the resolution derivation is in small space, one constructs a satisfying assignment for each clause configuration. Such assignments are updated inductively as the derivation progresses, and one shows that the update is always possible given the assumption that the space is small. This in turn shows that the contradictory empty clause can never be reached, implying a space lower bound on refutations. The essential feature separating this kind of proofs from the ones above is that the satisfying assignments arising during the proof *depend on the history of the derivation*; in contrast, the complexity measures in width lower bounds are defined once and for all, as are the distributions of random restrictions in length lower bounds.

In this section we present a *static* lower bound on resolution space. Our proof combines the ideas of Section 3 and the complexity measure for clauses used in [10]. We define a complexity measure for configurations which can be used to prove space lower bounds along the lines of the width lower bounds mentioned above.

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

▶ **Definition 7** (Tseitin formula). Let $G = (V, E)$ be an undirected graph and $\chi\colon V \to \{0, 1\}$ be a function. Identify every edge $e \in E$ with a variable x_e, and let $PARITY_{v,\chi}$ denote the CNF encoding of the constraint $\sum_{e \ni v} x_e = \chi(v) \pmod 2$ for any vertex $v \in V$. Then the *Tseitin formula* over G with respect to χ is $\mathit{Ts}(G, \chi) = \bigwedge_{v \in V} PARITY_{v,\chi}$.

When the degree of G is bounded by d, $PARITY_{v,\chi}$ has at most 2^{d-1} clauses, all of width at most d, and hence $\mathit{Ts}(G, \chi)$ is a d-CNF formula with at most $2^{d-1}|V|$ clauses. We say that a set of vertices U has *odd (even) charge* if $\sum_{u \in U} \chi(u)$ is odd (even). A simple counting argument shows that when $V(G)$ has odd charge, $\mathit{Ts}(G, \chi)$ is unsatisfiable. On the other hand, if G is connected then for each $v \in V$ it is always possible to satisfy the constraints $PARITY_{u,\chi}$ for all $u \neq v$. If G is a good expander, then large space is needed to refute $\mathit{Ts}(G, \chi)$.

▶ **Definition 8** (Edge expansion). The graph $G = (V, E)$ is an (s, δ)-*edge expander* if for every set of vertices $U \subseteq V$ such that $|U| \leq s$, the set of edges $E\bigl(U\bigr)$ has size at least $\delta|U|$, where $E\bigl(U\bigr)$ is the set of edges of G with exactly one vertex in U.

▶ **Theorem 9.** *For a d-degree (s, δ)-edge expander G it holds that $Sp(\mathit{Ts}(G, \chi)) \geq \delta s / d$.*

We remark that Theorem 9 was originally proven in [1, 13] (and with slightly better parameters, as discussed below).

For the rest of this section we fix a particular connected graph G of degree d, a function χ with respect to which $V(G)$ has odd charge, and the corresponding Tseitin formula $\mathit{Ts}(G, \chi)$. The main tool used to prove Theorem 9 is a complexity measure for configurations. We show that if G is a good expander, then every refutation of $\mathit{Ts}(G, \chi)$ must have a configuration with intermediate measure. We conclude the proof by showing that the space of a configuration is at least the value of its measure, if the latter falls within a specific range of values.

We first define our configuration complexity measure for terms (i.e. configurations consisting of unit clauses), and then we extend it to general configurations. In words, the term complexity measure is the smallest number of parity axioms of $\mathit{Ts}(G, \chi)$ that collectively contradict the term, and the configuration complexity measure is the maximum measure over all terms that imply the configuration.

▶ **Definition 10** (Configuration complexity measure). The *term complexity measure* $\nu(T)$ of a term T is $\nu(T) = \min\bigl\{|V'| : V' \subseteq V \text{ and } T \wedge \bigwedge_{v \in V'} PARITY_{v,\chi} \vDash \bot\bigr\}$.

The *configuration complexity measure* $\mu(\mathbb{C})$ of a resolution configuration $\mathbb{C}$ is defined as $\mu(\mathbb{C}) = \max\{\nu(T) : T \vDash \mathbb{C}\}$. When only trivial terms T imply $\mathbb{C}$, we have $\mu(\mathbb{C}) = 0$.

We now introduce the convenient concept of *witness* for the measure: a witness for $\nu(T)$ is a set of vertices V^* for which $\nu(T) = |V^*|$ and $T \wedge \bigwedge_{v \in V^*} \mathit{PARITY}_{v,\chi} \vDash \bot$. Similarly, for configurations, a witness for $\mu(\mathbb{C})$ is a term T^* for which $\nu(T^*) = \mu(\mathbb{C})$ and $T^* \vDash \mathbb{C}$.

There is a big gap between the measure of the initial and final configurations of a refutation, and we will see that the measure does not change much at each step. Hence, the refutation must pass through a configuration of intermediate measure. Formally, we have that $\mu(\emptyset) = |V|$, because the empty term implies $\emptyset$ and has measure $|V|$, and $\mu(\mathbb{C}) = 0$ when $\bot \in \mathbb{C}$, as only trivial terms imply contradiction.

To study how the measure changes during the refutation, we look separately at what happens at each type of step. As in the proof of Theorem 1, we can deal with inference and clause erasure steps together.

▶ **Lemma 11.** *If $\mathbb{C} \vDash \mathbb{C}'$ then $\mu(\mathbb{C}) \leq \mu(\mathbb{C}')$.*

Proof. Let T^* be a witness for $\mu(\mathbb{C})$. Then, $T^* \vDash \mathbb{C}$ and, hence, we also have $T^* \vDash \mathbb{C}'$. Therefore, $\mu(\mathbb{C}') \geq \nu(T^*) = \mu(\mathbb{C})$. ◀

Again, as in the proof of Theorem 1, axiom download requires most of the work. We show that if the graph has constant degree d, then the measure decreases slowly.

▶ **Lemma 12.** *For a clause A in $\mathit{Ts}(G,\chi)$ and a graph G of bounded degree d, if $\mathbb{C}' = \mathbb{C} \cup \{A\}$ then $d \cdot \mu(\mathbb{C}') + 1 \geq \mu(\mathbb{C})$.*

Proof. Fix a witness T^* for $\mu(\mathbb{C})$. Since $\mu(\mathbb{C}) = \nu(T^*)$, to prove the lemma we need to upper-bound the value $\nu(T^*)$ by $d \cdot \mu(\mathbb{C}') + 1$.

For any literal a in A, we know that $T^* \wedge a$ implies $\mathbb{C}'$ because T^* implies $\mathbb{C}$ and a implies A. Hence, it holds that $\mu(\mathbb{C}') \geq \nu(T^* \wedge a)$, and so it will be sufficient to relate $\nu(T^*)$ to the values $\nu(T^* \wedge a)$. To this end, we look at the set of vertices $V^* = \bigcup_{a \in A} V_a \cup \{v_A\}$, where each V_a is a witness for the corresponding measure $\nu(T^* \wedge a)$, and v_A is the vertex such that $A \in \mathit{PARITY}_{v_A,\chi}$. Note that by definition we have $|V_a| = \nu(T^* \wedge a)$ for every $a \in A$ and also that $|V^*| \leq \sum_{a \in A} |V_a| + 1$, which can in turn be bounded by $d \cdot \mu(\mathbb{C}') + 1$ because A has at most d literals.

We conclude the proof by showing that $T^* \wedge \bigwedge_{v \in V^*} \mathit{PARITY}_{v,\chi} \vDash \bot$, which shows that $\nu(T^*) \leq |V^*|$. The implication holds because any assignment either falsifies clause A, and so falsifies $\mathit{PARITY}_{v_A,\chi}$, or one of the literals $a \in A$ is satisfied. But then we have as a subformula $T^* \wedge \bigwedge_{v \in V_a} \mathit{PARITY}_{v,\chi}$, which is unsatisfiable by the definition of V_a when a is true. The bound $\nu(T^*) \leq |V^*|$ then follows, and so $\mu(\mathbb{C}) \leq |V^*| \leq d \cdot \mu(\mathbb{C}') + 1$. ◀

The preceding results imply that every resolution refutation of the Tseitin formula has a configuration of intermediate complexity. This holds because every refutation starts with a configuration of measure $|V|$ and needs to reach the configuration of measure 0, while at each step the measure drops at most a factor $1/d$ by previous lemmas.

▶ **Corollary 13.** *For any resolution refutation π of a Tseitin formula $\mathit{Ts}(G,\chi)$ over a connected graph G of bounded degree d and any positive integer $r \leq |V|$ there exists a configuration $\mathbb{C} \in \pi$ such that the configuration complexity measure is bounded by $r/d \leq \mu(\mathbb{C}) \leq r$.*

It remains to show that a configuration having intermediate measure must also have large space. Note that $\nu(T)$ is a monotone decreasing function, since $T \subseteq T'$ implies $\nu(T) \geq \nu(T')$

by definition. Hence, we only need to look at minimal terms T for which $T \vDash \mathbb{C}$ in order to determine $\mu(\mathbb{C})$.

In the case of expander graphs we have a space lower bound from the configuration complexity measure.

▶ **Lemma 14.** *Let G be an (s, δ)-edge expander graph. For every configuration $\mathbb{C}$ satisfying $\mu(\mathbb{C}) \leq s$ it holds that $Sp(\mathbb{C}) \geq \delta \cdot \mu(\mathbb{C})$.*

Proof. To prove the lemma, we lower-bound the size of a minimal witness T^* for $\mu(\mathbb{C})$ and then use the bound $Sp(\mathbb{C}) \geq |T^*|$. If only trivial terms imply $\mathbb{C}$ then the lemma immediately follows because $\mu(\mathbb{C}) = 0$. The latter bound follows by noting that every literal of T^* must imply at least one clause in $\mathbb{C}$. Fix T^* to be a minimal witness for $\mu(\mathbb{C})$ and let V^* be a witness for $\nu(T^*)$. Note that $|V^*| = \mu(\mathbb{C})$. We prove that T^* must contain a variable for every edge in $E\bigl(V^*\bigr)$.

Towards contradiction, assume that T^* does not contain some x_e for an edge e in $E\bigl(V^*\bigr)$, and let v_e be the vertex in V^* incident to e. Let α be an assignment that satisfies $T^* \wedge \bigwedge_{v \in V^* \setminus \{v_e\}} PARITY_{v,\chi}$. Such an assignment must exist as otherwise V^* would not be a witness for $\nu(T^*)$. We can modify α by changing the value of x_e so that $PARITY_{v_e,\chi}$ is satisfied. By the assumption, the new assignment α' still satisfies T^* and $\bigwedge_{v \in V^* \setminus \{v_e\}} PARITY_{v,\chi}$ as neither contains the variable x_e. Thus, we have found an assignment satisfying $T^* \wedge \bigwedge_{v \in V^*} PARITY_{v,\chi}$, which is a contradiction.

Hence, the term T^* contains a variable for every edge in $E\bigl(V^*\bigr)$. Since G is an (s, δ)-edge expander and $|V^*| \leq s$, the term T^* contains at least $\delta \cdot |V^*|$ variables. From $Sp(\mathbb{C}) \geq |T^*|$ and the fact that $|V^*| = \mu(\mathbb{C})$ we prove that $Sp(\mathbb{C}) \geq \delta \cdot \mu(\mathbb{C})$ if $\mu(\mathbb{C}) \leq s$. ◀

The preceding lemma and Corollary 13 together imply Theorem 9, because by Corollary 13 there is a configuration with measure between s/d and s, and this configuration has space at least $\delta s/d$ by the previous lemma.

Theorem 9 gives inferior results compared to a direct application of Theorem 1 to known width lower bounds. The bounds that we get are worse by a multiplicative factor of $1/d$. One might hope to remove this multiplicative factor by improving the bound in Lemma 12, but this is not possible because that bound is tight.

To see this, assume that the graph G consists of a set of vertices V with one vertex v that is a neighbor of d disjoint subgraphs each of size $(|V| - 1)/d$. Also, let A be one of the clauses in $PARITY_{v,\chi}$ such that setting any literal in A to true pushes the odd charge into one of the neighboring subgraphs of v. Taking $\mathbb{C} = \emptyset$ and $\mathbb{C}' = \{A\}$ we have that $\mu(\mathbb{C}) = |V|$ and $\mu(\mathbb{C}') = (|V| - 1)/d$. The latter equality holds because every minimal term T satisfying A contains exactly one literal from A, and so pushes the odd charge into one of the subgraphs neighboring v. This makes the vertices of that subgraph a witness for $\nu(T)$. Hence, we have an example where $d \cdot \mu(\mathbb{C}') + 1 = \mu(\mathbb{C})$, which shows that Lemma 12 is tight.

5 From Small Space to Small Degree in Polynomial Calculus?

An intriguing question is whether an analogue of the bound in Theorem 1 holds also for the stronger algebraic proof system *polynomial calculus* introduced in [12]. In this context, it is more relevant to discuss the variant of this system presented in [1], which is known as *polynomial calculus (with) resolution* or *PCR*, which we briefly describe below.

In a PCR derivation, the configurations are sets of polynomials in $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$, where x and $\overline{x}$ are different formal variables. By way of example, a clause $x \vee y \vee \overline{z}$ is translated to the polynomial $xy\overline{z}$ consisting of one monomial. In addition to the translations of the axiom clauses of the CNF formula to be refuted, the proof system also contain axioms $x^2 - x$ and $x + \overline{x} - 1$. These axioms enforce that only assigments to $\{0, 1\}$ are considered (and hence that all polynomials are multilinear without loss of generality) and that $\overline{x}$ always takes the opposite value of x. There are two inference rules, which preserve common roots of the polynomials, namely *linear combination* $\frac{p \quad q}{\alpha p + \beta q}$ and *multiplication* $\frac{p}{xp}$, where p and q are (previously derived) polynomials, the coefficients α, β are elements of $\mathbb{F}$, and x is any variable (with or without bar). The refutation ends when 1 has been derived. The *size*, *degree* and *monomial space* measures are analogues of length, width and clause space in resolution (counting monomials instead of clauses). PCR can simulate resolution refutations efficiently with respect to all of these measures.

Let us now discuss why the method we use to prove Theorem 1 is unlikely to generalize to PCR. An example of formulas that seem hard to deal with in this way are so-called *pebbling contradictions*, which we describe next.

Pebbling contradictions are defined in terms of directed acyclic graphs (DAGs) $G = (V, E)$ with bounded fan-in, where vertices with no incoming edges are called *sources* and vertices without outgoing edges *sinks*. Assume G has a unique sink z and associate a variable V to each vertex $v \in V$. Then the pebbling contradiction over G consists of the following clauses:

- for each source vertex s, a clause s (*source axioms*),
- for each non-source vertex v, a clause $\bigvee_{(u,v)\in E} \overline{u} \vee v$ (*pebbling axioms*),
- for the sink z, a clause $\overline{z}$ (*sink axiom*).

As shown in [6], pebbling contradictions exhibit space-width trade-offs in resolution in that they can always be refuted in constant width as well as in constant space, but there are graphs for which optimizing one of these measures necessary causes essentially worst-case linear behaviour for the other measure.

There are two natural ways to refute pebbling contradictions in resolution. One approach is to go "bottom-up" from sources to sinks in topological order, and derive for each vertex $v \in V(G)$ the clause v using the pebbling axiom for v and the clauses for the predecessors of the vertex v. When the refutation reaches z it derives a contradiction with the sink axiom $\overline{z}$. This can be done in constant width but for some graphs requires large space. The other approach is "top-down" starting from the sink axiom $\overline{z}$ and deriving clauses of the form $\overline{v}_1 \vee \cdots \vee \overline{v}_\ell$. A new clause is derived by replacing any vertex v_i in the old one by all its predecessors, i.e., resolving with the pebbling axiom for v_i. Since G is acyclic we can repeat this process until we get to the sources, for which the negated literals can be resolved away using source axioms. This refutation can be carried out in constant clause space, but such a refutation might require large width.

Now, one can observe that the transformation of configurations in our proof of Theorem 1 maps either of two refutations above into the other one, and this is the main reason why our proof does not seem to generalize to PCR. In PCR, we can represent any conjunction of literals $a_1 \wedge \cdots \wedge a_r$ as the binomial $1 - \prod_i \overline{a}_i$. Using this encoding with the bottom-up approach yields a third refutation, which has constant space but possibly large degree. Hence, there are constant space polynomial calculus refutations of pebbling contradictions in both the bottom-up and the top-down direction. This in turn means that if our proof method were to work for PCR, we would need to find constant degree refutations in both directions. For the top-down case it seems unlikely that such a refutation exists.

6 Concluding Remarks

In this work, we present an alternative, completely elementary, proof of the result by Atserias and Dalmau [2] that space is an upper bound on width in resolution. Our construction gives a syntactic way to convert a small-space resolution refutation into a refutation in small width. We also exhibit a new "black-box" approach for proving space lower bounds that works by defining a progress measure à la Ben-Sasson and Wigderson [10] and showing that when a refutation has made medium progress towards a contradiction it must be using a lot of space. We believe that these techniques shed interesting new light on resolution space complexity, and hope that they will serve to increase our understanding of this notoriously tricky complexity measure.

As an example of a question about resolution space that still remains open, suppose we are given a k-CNF formula that is guaranteed to be refutable in constant space. By [2] it is also refutable in constant width, and a simple counting argument then shows that exhaustive search in small width will find a polynomial-length resolution refutation. But is there any way of obtaining such a short refutation from a refutation in small space that is more explicit that doing exhaustive search? And can we obtain a short refutation without blowing up the space by more than, say, a constant factor?

Known length-space trade-off results for resolution in [4, 5, 9, 17] do not answer this question as they do not apply to this range of parameters. Unfortunately, our new proof of the space-width inequality cannot be used to resolve this question either, since in the worst case the resolution refutation we obtain might be as bad as the one found by exhaustive search of small-width refutations (or even worse, due to repetition of clauses). This would seem to be inherent—a recent result [3] shows that there are formulas refutable in space and width s where the shortest refutation has length $n^{\Omega(s)}$, i.e., matching the exhaustive search upper bound up to a (small) constant factor in the exponent.

An even more intriguing question is how the space and degree measures are related in polynomial calculus, as discussed in Section 5. For most relations between length, space, and width in resolution, it turns out that they carry over with little or no modification to size, space, and degree, respectively, in polynomial calculus. So can it be that it also holds that space yields upper bounds on degree in polynomial calculus? Or could perhaps even the stronger claim hold that polynomial calculus space is an upper bound on resolution width? These questions remain wide open, but in the recent paper [14] we made some limited progress by showing that if a formula requires large resolution width, then the "XORified version" of the formula requires large polynomial calculus space. We refer to the introductory section of [14] for a more detailed discussion of these issues.

Acknowledgments. The authors wish to thank Albert Atserias, Ilario Bonacina, Nicola Galesi, and Li-Yang Tan for stimulating discussions on topics related to this work.

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

References

1 Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version appeared in *STOC '00*.

2 Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, May 2008. Preliminary version appeared in *CCC '03*.

3 Albert Atserias, Massimo Lauria, and Jakob Nordström. Narrow proofs may be maximally long. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC'14)*, to appear, 2014.

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

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

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

7 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*.

8 Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proc. of the 49th Annual IEEE Symp. on Foundations of Computer Science (FOCS '08)*, pages 709–718, October 2008.

9 Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proc. of the 2nd Symp. 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/`.

10 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*.

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

12 Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proc. of the 28th Annual ACM Symp. on Theory of Computing (STOC '96)*, pages 174–183, May 1996.

13 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*.

14 Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (extended abstract). In *Proc. of the 40th Int'l Colloquium on Automata, Languages and Programming (ICALP '13)*, volume 7965 of *LNCS*, pages 437–448. Springer, July 2013.

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

16 Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1–3):123–140, 2001.

17 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.

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

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