From Small Space to Small Width in Resolution

YUVAL FILMUS, Institute for Advanced Study
MASSIMO LAURIA, MLADEN MIKŠA, JAKOB NORDSTRÖM, and MARC VINYALS, KTH
Royal Institute of Technology

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

Categories and Subject Descriptors: F.2.2 [**Theory of Computation**]: Proof Complexity; G.2.1 [**Mathematics of Computing**]: Combinatoric Problems

Additional Key Words and Phrases: Proof complexity, resolution, width, space, polynomial calculus, polynomial calculus resolution, PCR, degree

ACM Reference Format:
Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. 2015. From small space to small width in resolution. ACM Trans. Comput. Logic 16, 4, Article 28 (July 2015), 15 pages.
DOI: http://dx.doi.org/10.1145/2746339

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.

This is a revised and expanded version of the paper Filmus et al. [2014], which appeared in *Proceedings of the 31st Symposium on Theoretical Aspects of Computer Science (STACS'14)*.
The research of the first author received funding from the European Union's Seventh Framework Programme (FP7/2007–2013) under grant agreement no. 238381 and from the National Science Foundation under agreement no. DMS-1128155. Part of the work of the first author was performed while at the University of Toronto and while visiting KTH Royal Institute of Technology supported by the foundations *Johan och Jakob Söderbergs stiftelse*, *Stiftelsen Längmanska kulturfonden*, and *Helge Ax:son Johnsons stiftelse*. The other authors were funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013)/ERC grant agreement no. 279611. The fourth author was also supported by Swedish Research Council grants 621-2010-4797 and 621-2012-5645. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors, and do not necessarily reflect the views of the National Science Foundation or other funding agencies.

Authors' addresses: Y. Filmus, Institute for Advanced Study, School of Mathematics, 1 Einstein Drive, Princeton, NJ, USA; email: yfilmus@ias.edu; M. Lauria, M. Mikša, J. Nordström, and M. Vinyals, KTH Royal Institute of Technology, School of Computer Science and Communication, SE-100 44 Stockholm, Sweden; emails: {lauria, miksa, jakobn, vinyals}@kth.se.

DOI: http://dx.doi.org/10.1145/2746339

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*, that is, the size of a largest clause in the proof. In a celebrated result, Ben-Sasson and Wigderson [2001] showed that there are short resolution refutations of a formula if and only if there are also (reasonably) narrow ones, and almost all known lower bounds on resolution length can be (re)derived using this connection. In 2003, Atserias and Dalmau (journal version in Atserias and Dalmau [2008]) established that width also provides lower bounds on space, resolving a problem that had been open since the study of space complexity of propositional proofs was initiated in the late 1990s in Alekhnovich et al. [2002] and Esteban and Torán [2001]. This means that for space also, almost all known lower bounds can be rederived by using width lower bounds and appealing to Atserias and Dalmau [2008]. 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 Ben-Sasson and Nordström [2008].

1.1. Our Contributions

The starting point of our work is the lower bound on space in terms of width in Atserias and Dalmau [2008]. This is a very elegant but also indirect proof in that it translates the whole problem to Ehrenfeucht–Fraïssé games in finite model theory, and shows that resolution space and width correspond to strategies for two opposite players in such games. Unfortunately, this also means that one obtains essentially no insight into what is happening on the proof complexity side (other than that the bound on space in terms of width is true). It has remained an open problem to give a more explicit, proof complexity theoretic argument.

In this article, 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 timestep in a small-space refutation, negate these conjunctions and then expand them to conjunctive normal form again, and finally argue that the new sets of clauses listed in reverse order (essentially) constitute a small-width refutation of the same formula.[1]

This new, simple proof also allows us to obtain a new technique for proving space lower bounds. This approach is reminiscent of Ben-Sasson and Wigderson [2001] 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 important motivation for our work was the hope that a simplified proof of the space-width inequality would serve as a stepping stone to resolving the analogous question for the polynomial calculus proof system. Here, the width of clauses corresponds to the *degree* of polynomials, space is measured as the total number of monomials of all

[1] We recently learned that a similar proof, though phrased in a slightly different language, was obtained independently by Razborov [2014].

polynomials currently in memory, and the problem is to determine whether space and degree in polynomial calculus are related in the same way as are space and width in resolution. A possible approach for attacking this question was proposed in Bonacina and Galesi [2013]. In Filmus et al. [2013] we obtained a result analogous to Ben-Sasson and Nordström [2008] that there are formulas of worst-case space complexity that require only constant degree. The question of whether degree lower bounds imply space lower bounds remains open, however, and other results in Filmus et al. [2013] can be interpreted as implying that the techniques in Bonacina and Galesi [2013] probably are not sufficient to resolve this question. Unfortunately, as discussed toward the end of this article, we also show that it appears unlikely that this problem can be addressed by methods similar to our proof of the corresponding inequality for resolution.

1.2. Outline of This Article

The rest of this article is organized as follows. After some brief preliminaries in Section 2, we present the new proof of the space-width inequality in Atserias and Dalmau [2008] in Section 3. In Section 4, we showcase the new technique for space lower bounds by studying so-called Tseitin formulas. Section 5 explains why we believe it is unlikely that our methods will extend to polynomial calculus. Some concluding remarks are given in Section 6.

2. PRELIMINARIES

Let us start with a brief review of the preliminaries. The following material is standard and can be found, for example, in the survey by Nordström [2013].

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

A clause C' *subsumes* another clause C if every literal from C' also appears in C. A *k-clause (k-term)* is a clause (term) that contains at most k literals. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses, and a *DNF formula* $F = T_1 \vee \cdots \vee T_m$ is a disjunction of terms. A *k-CNF formula (k-DNF formula)* is a CNF formula (DNF formula) consisting of k-clauses (k-terms). We think of clauses, terms, and CNF formulas as sets: the order of elements is irrelevant and there are no repetitions. We also assume that CNF formulas are nontrivial in the sense that they do not contain the contradictory empty clause (this is just for technical simplicity to avoid a pathological corner case).

Let us next describe a slight generalization of the resolution proof system due to Krajíček [2001], who introduced the family of *r-DNF resolution* proof systems, denoted $\mathcal{R}(r)$, as an intermediate step between resolution and depth-2 Frege systems. An *r-DNF resolution configuration* $\mathbb{C}$ is a set of r-DNF formulas. An *r-DNF resolution refutation* of a CNF formula F is a sequence of configurations $(\mathbb{C}_0, \ldots, \mathbb{C}_\tau)$ such that $\mathbb{C}_0 = \emptyset$, $\perp \in \mathbb{C}_\tau$, and for $1 \leq t \leq \tau$ we obtain $\mathbb{C}_t$ from $\mathbb{C}_{t-1}$ by one of the following steps:

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

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

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

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

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

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

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

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

$$\frac{C \vee x \quad D \vee \overline{x}}{C \vee D}. \tag{1}$$

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

The *length* $L(\pi)$ of an r-DNF resolution refutation π is the number of download and inference steps, and the *space* $Sp(\pi)$ is the maximal number of r-DNF formulas in any configuration in π. We define the length $L_{\mathcal{R}(r)}(F \vdash \perp)$ and the space $Sp_{\mathcal{R}(r)}(F \vdash \perp)$ 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 $W(F \vdash \perp)$ of refuting F. We remark that in the context of resolution the space measure defined previously is sometimes referred to as *clause space* to distinguish it from other space measures studied for this proof system.

3. FROM SPACE TO WIDTH

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

THEOREM 3.1 ([ATSERIAS AND DALMAU 2008]). *Let F be a k-CNF formula and let $\pi : F \vdash \perp$ be a resolution refutation in clause space $Sp(\pi) = s$. Then, there is a resolution refutation π' of F in width $W(\pi') \leq s + k - 3$.*

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

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

Definition 3.2. The *negated configuration* $\mathrm{neg}(\mathbb{C})$ of a clause configuration $\mathbb{C}$ is defined inductively as follows:

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

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

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

OBSERVATION 3.3. *The width of any clause in the negated configuration* $\mathrm{neg}(\mathbb{C})$ *is at most* $Sp(\mathbb{C}_t) = |\mathbb{C}|$.

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

PROPOSITION 3.4. *The negated configuration* $\mathrm{neg}(\mathbb{C})$ *is the set of all minimal (nontrivial) 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 a minimal clause $C' \subseteq C$ such that $\neg C' \vDash \mathbb{C}$ we have that $C' \in \mathbb{D}$.

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

The following observation, which formalizes the main idea behind the concept of negated configurations, is an immediate consequence of Proposition 3.4.

OBSERVATION 3.5. *An assignment satisfies a clause configuration $\mathbb{C}$ if and only if it falsifies the negated clause configuration* $\mathrm{neg}(\mathbb{C})$. *That is, $\mathbb{C}$ is logically equivalent to* $\neg\mathrm{neg}(\mathbb{C})$.

Recall that 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 complexity, then the reversed sequence of negated configurations $\pi' = (\mathrm{neg}(\mathbb{C}_\tau), \mathrm{neg}(\mathbb{C}_{\tau-1}), \ldots, \mathrm{neg}(\mathbb{C}_0))$ has small width complexity. However, as noted previously, 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, that is, axiom download, clause inference, and clause erasure. The following lemma shows that for inference and erasure steps all that is needed in the reverse direction is to apply weakening.

LEMMA 3.6. *If* $\mathbb{C} \vDash \mathbb{C}'$*, then for every clause* $C \in \mathrm{neg}(\mathbb{C})$ *there exists a clause* $C' \in \mathrm{neg}(\mathbb{C}')$ *such that* C *is a weakening of* C'*.*

PROOF. For any clause C in $\mathrm{neg}(\mathbb{C})$ it holds by Proposition 3.4 that $\neg C \vDash \mathbb{C}$. Since $\mathbb{C} \vDash \mathbb{C}'$, this in turns implies that $\neg C \vDash \mathbb{C}'$. Applying Proposition 3.4 again, we conclude that there exists a clause $C' \subseteq C$ such that $C' \in \mathrm{neg}(\mathbb{C}')$. □

The only time in a refutation $\pi = (\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau)$ when it does not hold that $\mathbb{C}_{t-1} \vDash \mathbb{C}_t$ is when an axiom clause is downloaded at time t, and such derivation steps will require a bit more careful analysis. We provide such an analysis in the full proof of Theorem 3.1, which we are now ready to present.

PROOF OF THEOREM 3.1. Let $\pi = (\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau)$ be a resolution refutation of F in space s. For every configuration $\mathbb{C}_t \in \pi$, let $\mathbb{D}_t = \mathrm{neg}(\mathbb{C}_t)$ denote the corresponding negated configuration. By assumption, each $\mathbb{C}_t$ contains at most s clauses, and thus Observation 3.3 guarantees that the clauses in $\mathbb{D}_t$ have width at most s. We need to show how to transform the sequence of clause configurations $\pi' = (\mathbb{D}_\tau, \mathbb{D}_{\tau-1}, \ldots, \mathbb{D}_0)$ into a legal resolution refutation of width at most $s+k-3$. Let us assume first that we are dealing with CNF formulas of width $k \geq 3$, since this makes the argument slightly easier to present. At the end of the proof, we will see how to argue more carefully to get rid of this assumption.

The initial configuration of the sequence π' is $\mathbb{D}_\tau$, which is the empty set by Definition 3.2. If $\mathbb{C}_{t+1}$ follows from $\mathbb{C}_t$ by inference or erasure, then we can derive any clause of $\mathbb{D}_t$ from a clause of $\mathbb{D}_{t+1}$ by weakening, as proven in Lemma 3.6. If $\mathbb{C}_{t+1}$ follows from $\mathbb{C}_t$ by axiom download, then we claim that we can derive $\mathbb{D}_t$ from $\mathbb{D}_{t+1}$ in width at most $s+k-3$. Since the last configuration $\mathbb{D}_0$ of π' contains the empty clause $\perp$ by Definition 3.2, we obtain a complete resolution refutation.

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

Suppose $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{A\}$ for some axiom $A = a_1 \vee \cdots \vee a_\ell$, with $\ell \leq k$. Consider now some clause $C \in \mathbb{D}_t \setminus \mathbb{D}_{t+1}$. By Observation 3.3 it holds that $W(C) \leq Sp(\mathbb{C}_t) \leq s-2$. To derive C we first download the axiom A and then show how to obtain C from the clauses in $\mathbb{D}_{t+1} \cup \{A\}$. Note that all clauses $C \vee \overline{a}_i$ for $a_i \in A$ are either contained in, or are weakenings of, clauses in $\mathbb{D}_{t+1}$. This follows easily from Definition 3.2 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:

$$\begin{array}{c}
\dfrac{\dfrac{\dfrac{A = a_1 \vee \cdots \vee a_\ell \qquad C \vee \overline{a}_1}{C \vee a_2 \vee \cdots \vee a_\ell} \qquad C \vee \overline{a}_2}{C \vee a_3 \vee \cdots \vee a_\ell}}{} \\
\vdots \\
\dfrac{C \vee a_\ell \qquad\qquad C \vee \overline{a}_\ell}{C}
\end{array} \tag{2}$$

When C is the empty clause, the width of this derivation is $W(A) \leq k$. Otherwise, it is upper bounded by $W(C) + W(A) - 1 \leq s+k-3$. Since any resolution refutation has space at least 3 (unless the formula contains the empty clause itself, but our definitions explicitly disallowed such trivial formulas), we conclude that the width of the derivation (2) is at most $\max(k, s + k - 3) = s + k - 3$. This in turn implies that the width of the resolution refutation constructed from π' is at most $\max(s, s + k - 3) = s + k - 3$, where the last equality follows from the assumption $k \geq 3$, and this completes the proof.

If $k < 3$, however, we have $s + k - 3 < s$, and so the preceding argument does not quite suffice to establish the bound claimed in the theorem. This can be taken care of by a postprocessing step as follows.[2] Recall that inference and erasure steps can only produce weakenings of clauses by Lemma 3.6, and axiom download steps only occur when the space is at most $s - 2$. Consider the resolution refutation constructed from π' as previously, and then erase all clause configurations obtained at inference or erasure steps (i.e., via weakening) to obtain new refutation π''. It is straightforward to verify that this yields a legal refutation and that the width does not increase (since π'' contains a subset of the clauses in the previously constructed refutation). After this step, the only new clauses in π'' that we need to derive at each step are those resulting from axiom downloads in the original refutation π, and as already noted the width of deriving such clauses as done in (2) is at most $\max(k, s + k - 3) = s + k - 3$. The theorem follows. □

The proof of Theorem 3.1 also works for r-DNF resolution, although the bound gets weaker as r grows. Let us state this as a theorem and sketch the proof.

THEOREM 3.7. *Let F be a k-CNF formula and $\pi : F \vdash \bot$ be an r-DNF resolution refutation of F in space $Sp(\pi) \leq s$. Then there exists a resolution refutation π' of F in width at most $W(\pi') \leq (s - 2)r + k - 1$.*

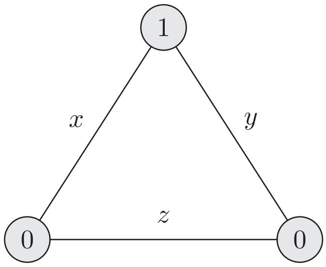
PROOF SKETCH. We define the negated configuration $neg_{\mathcal{R}(r)}(\mathbb{C})$ of an $\mathcal{R}(r)$-configuration inductively by setting $neg_{\mathcal{R}(r)}(\emptyset) = \{\bot\}$ and

$$
\begin{aligned}
&neg_{\mathcal{R}(r)}(\mathbb{C} \cup \{C\}) \\
&\quad = \{D \vee \overline{T} \mid D \in neg_{\mathcal{R}(r)}(\mathbb{C}); T \in C; \nexists B \in neg_{\mathcal{R}(r)}(\mathbb{C}) \text{ s.t. } B \vee \overline{T} \subsetneq D \vee \overline{T}; D \vee \overline{T} \text{ nontrivial}\} \quad (3)
\end{aligned}
$$

to make sure that $neg_{\mathcal{R}(r)}(\mathbb{C})$ contains no trivial or subsumed clauses. It is easy to see that an r-DNF configuration of space s gets transformed into a resolution configuration of width at most sr. We can prove that $neg_{\mathcal{R}(r)}(\mathbb{C})$ is the set of all minimal clauses D such that $\neg D \vDash \mathbb{C}$ for an r-DNF configuration $\mathbb{C}$, which is an analog of Proposition 3.4. The proof is essentially the same except that we reason using the terms of an r-DNF formula $C \in \mathbb{C}$ instead of its literals. With this version of Proposition 3.4 proved, we can immediately generalize Lemma 3.6 to the r-DNF case.

The analog of the proof of Theorem 3.1 follows easily from previous observations. The inference and clause deletion steps follow by the generalized version of Lemma 3.6, while the case of axiom download is the same as in the original proof because axioms are clauses. Hence, running the negated r-DNF resolution refutation backward we get a resolution refutation of F. The width of this latter refutation is at most $(s-2)r + k-1$, as we again consider only configurations that have space equal to at most $s - 2$, and the inference steps in the case of axiom download can add at most $k - 1$ to the width

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



(a) Labeled triangle graph.

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

(b) Corresponding Tseitin formula.

Fig. 1. Example Tseitin formula.

of the resulting resolution refutation. When $k < 2r + 1$, an additional pruning step in which all weakenings are eliminated completes the proof. □

4. A STATIC TECHNIQUE FOR PROVING SPACE LOWER BOUNDS

Looking at the proof complexity literature, the techniques used to prove lower bounds for resolution length and width (e.g., Ben-Sasson and Wigderson [2001], Chvátal and Szemerédi [1988], Haken [1985], and Urquhart [1987]) differ significantly from those used to prove resolution space lower bounds (e.g., Alekhnovich et al. [2002], Ben-Sasson and Galesi [2003], and Esteban and Torán [2001]) in that the former are *static* or *oblivious*, while the latter are *dynamic*.

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

In contrast, most lower bound proofs for resolution space use an *adversary argument*. Assuming that the resolution derivation has small space, one constructs a satisfying assignment for each clause configuration. Such assignments are updated inductively as the derivation progresses, and one shows that the update is always possible given the assumption that the space is small. This in turn shows that the contradictory empty clause can never be reached, implying a space lower bound on refutations. The essential feature separating this kind of proof from the previous ones 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 Ben-Sasson and Wigderson [2001]. We define a complexity measure for configurations that can be used to prove space lower bounds along the lines of the width lower bounds mentioned previously.

This approach works, in general, in that any complexity measure for clauses can be transformed into a complexity measure for configurations. This turns many width lower bound techniques into space lower bound ones (e.g., width lower bounds for random 3-CNF formulas). In this section, we give a concrete example of this for Tseitin formulas, which are a family of CNFs encoding a specific type of systems of linear equations (see Figure 1 for illustration).

Definition 4.1 (*Tseitin formula*). Let $G = (V, E)$ be an undirected graph and $\chi : V \to \{0, 1\}$ be a function. Let us identify every edge $e \in E$ with a variable x_e, and let us write $PARITY_{v,\chi}$ to denote the canonical CNF encoding of the constraint $\sum_{e \ni v} x_e = \chi(v) \pmod 2$ for any vertex $v \in V$. Then the *Tseitin formula* over G with respect to χ is $Ts(G, \chi) = \bigwedge_{v \in V} PARITY_{v,\chi}$.

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

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

Definition 4.2 (*Edge expander*). The graph $G = (V, E)$ is an (s, δ)-*edge expander* if for every set of vertices $U \subseteq V$ such that $|U| \leq s$ it holds that $|\partial(U)| \geq \delta|U|$, where $\partial(U)$ is the set of edges of G with exactly one vertex in U.

We next present a new technique of showing that if a graph G is a good edge expander, then large space is needed to refute $Ts(G, \chi)$ in resolution. We remark that this was originally proven in Alekhnovich et al. [2002] and Esteban and Torán [2001] (and with slightly better parameters, as discussed in the following).

THEOREM 4.3. *For a Tseitin formula $Ts(G, \chi)$ over a d-regular (s, δ)-edge expander G it holds that $Sp(Ts(G, \chi) \vdash \bot) \geq \delta s/d$.*

For the rest of this section we fix a particular d-regular connected graph G and a function χ with respect to which $V(G)$ has odd charge, and consider the corresponding Tseitin formula $Ts(G, \chi)$. The main tool used to prove Theorem 4.3 is a complexity measure for configurations. We show that if G is a good expander, then every refutation of $Ts(G, \chi)$ must have a configuration with intermediate measure. We conclude the proof by showing that the space of a configuration is at least its measure if the latter falls within a specific range of values.

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

Definition 4.4 (*Configuration complexity measure*). The *term complexity measure* $\nu(T)$ of a term T is $\nu(T) = \min\{|V'| : V' \subseteq V \text{ and } T \wedge \bigwedge_{v \in V'} PARITY_{v,\chi} \vDash \bot\}$.

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

Note that $\nu(T)$ is a monotone decreasing function, since $T \subseteq T'$ implies $\nu(T) \geq \nu(T')$ by definition. Hence, we only need to look at minimal terms T for which $T \vDash \mathbb{C}$ in order to determine $\mu(\mathbb{C})$. These minimal terms are the *negations* of the clauses in $\mathrm{neg}(\mathbb{C})$ (compare Proposition 3.4). We now introduce the convenient concept of *witness* for the measure.

Definition 4.5 (*Witness of measure*). A *witness* of the measure $\nu(T)$ of the term T is a set of vertices V^* for which $\nu(T) = |V^*|$ and $T \wedge \bigwedge_{v \in V^*} PARITY_{v,\chi} \vDash \bot$. Similarly, for configurations $\mathbb{C}$ a witness for $\mu(\mathbb{C})$ is a term T^* for which $\mu(\mathbb{C}) = \nu(T^*)$ and $T^* \vDash \mathbb{C}$.

There is a big gap between the measure of the initial and final configurations of a refutation, and we will see that the measure does not change much at each step. Hence, the refutation must pass through a configuration of intermediate measure. Formally, if G is connected, then $\mu(\emptyset) = |V|$, because the empty term has measure $|V|$, and $\mu(\mathbb{C}) = 0$ when $\perp \in \mathbb{C}$.

To study how the measure changes during the refutation, we look separately at what happens at each type of step. As in the proof of Theorem 3.1, we can deal with inference and clause erasure steps together, whereas axiom downloads require more work.

Lemma 4.6. *If $\mathbb{C} \vDash \mathbb{C}'$, then $\mu(\mathbb{C}) \leq \mu(\mathbb{C}')$.*

Proof. Let T^* be a witness for $\mu(\mathbb{C})$. Then, $T^* \vDash \mathbb{C}$ and, hence, we also have $T^* \vDash \mathbb{C}'$. Therefore, $\mu(\mathbb{C}') \geq \nu(T^*)$, because $\mu(\mathbb{C}')$ is equal to the maximum value of $\nu(T)$ for terms T implying $\mathbb{C}'$. As $\nu(T^*)$ is equal to $\mu(\mathbb{C})$, the bound $\mu(\mathbb{C}') \geq \mu(\mathbb{C})$ follows. □

Lemma 4.7. *For a clause A in $Ts(G, \chi)$ and a graph G of bounded degree d, if $\mathbb{C}' = \mathbb{C} \cup \{A\}$, then $d \cdot \mu(\mathbb{C}') + 1 \geq \mu(\mathbb{C})$.*

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

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

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

The preceding results imply that every resolution refutation of the Tseitin formula has a configuration of intermediate complexity. This holds because every refutation starts with a configuration of measure $|V|$ and needs to reach a configuration of measure 0, as noted previously, while at each step the measure drops by a factor of at most $1/d$ by the lemmas we just proved. Let us state this formally as a corollary.

Corollary 4.8. *For any resolution refutation π of a Tseitin formula $Ts(G, \chi)$ over a connected graph G of bounded degree d and any positive integer $r \leq |V|$ there exists a configuration $\mathbb{C} \in \pi$ such that the configuration complexity measure is bounded by $r/d \leq \mu(\mathbb{C}) \leq r$.*

It remains to show that a configuration having intermediate measure must also have large space. This part of the proof relies on the graph being an expander.

Lemma 4.9. *Let G be an (s, δ)-edge expander graph. For every configuration $\mathbb{C}$ satisfying $\mu(\mathbb{C}) \leq s$ it holds that $Sp(\mathbb{C}) \geq \delta \cdot \mu(\mathbb{C})$.*

Proof. To prove the lemma, we lower bound the size of a minimal witness T^* for $\mu(\mathbb{C})$ and then use the bound $Sp(\mathbb{C}) \geq |T^*|$. This inequality follows by noting that at most one literal per clause in $\mathbb{C}$ is needed in the implying term T^*.

Fix T^* to be a minimal witness for $\mu(\mathbb{C})$ and let V^* be a witness for $\nu(T^*)$. Note that $|V^*| = \mu(\mathbb{C})$. We prove that T^* must contain a variable for every edge in $\partial(V^*)$.

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

Hence, the term T^* contains a variable for every edge in $\partial(V^*)$. Since G is an (s, δ)-edge expander and $|V^*| \leq s$, the term T^* contains at least $\delta \cdot |V^*|$ variables. From the inequality $Sp(\mathbb{C}) \geq |T^*|$ and the fact that $|V^*| = \mu(\mathbb{C})$ it follows that $Sp(\mathbb{C}) \geq \delta \cdot \mu(\mathbb{C})$ when $\mu(\mathbb{C}) \leq s$. □

The preceding lemma and Corollary 4.8 together imply Theorem 4.3, because by Corollary 4.8 there is a configuration with measure between s/d and s, and this configuration has space at least $\delta s/d$ by Lemma 4.9.

We want to point out that Theorem 4.3 gives inferior results compared to a direct application of Theorem 3.1 to known width lower bounds. The bounds that we get are worse by a multiplicative factor of $1/d$. One might have hoped to remove this multiplicative factor by improving the bound in Lemma 4.7, but this is not possible because this lemma is tight.

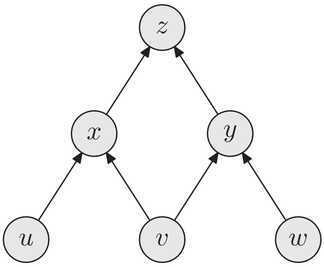
To see this, suppose that the graph G is a d-star: it consists of a center v which is connected to d petals $u_1, \ldots, u_d$ by the edges $e_1, \ldots, e_d$, the charge of the center is $\chi(v) = 1$, and the charges of the petals are $\chi(u_1) = \cdots = \chi(u_d) = 0$. Let $A \in PARITY_{v,\chi}$ be the axiom $A = x_{e_1} \vee \cdots \vee x_{e_d}$. Taking $\mathbb{C} = \emptyset$ and $\mathbb{C}' = \{A\}$, we have that $\mu(\mathbb{C}) = d + 1$, while $\mu(\mathbb{C}') = 1$. The latter equality holds because every minimal term implying A is of the form x_{e_i}, a term which is contradicted by the single axiom $\overline{x}_{e_i} \in PARITY_{u_i,\chi}$. Hence, we have an example where $d \cdot \mu(\mathbb{C}') + 1 = \mu(\mathbb{C})$, which shows that Lemma 4.7 is tight.

5. FROM SMALL SPACE TO SMALL DEGREE IN POLYNOMIAL CALCULUS?

An intriguing question is whether an analog of the bound in Theorem 3.1 holds also for the stronger algebraic proof system *polynomial calculus* introduced in Clegg et al. [1996]. In this context, it is more relevant to discuss the variant of this system presented in Alekhnovich et al. [2002], known as *Polynomial Calculus (with) Resolution* or *PCR*, which we briefly describe next.

In a PCR derivation, configurations are sets of polynomials in $\mathbb{F}[x, \overline{x}, y, \overline{y}, \ldots]$, where x and $\overline{x}$ are different formal variables. Each polynomial P appearing in a configuration corresponds to the assertion $P = 0$. The proof system contains axioms $x^2 - x$ and $x + \overline{x} - 1$, which restrict the values of the variables to $\{0, 1\}$, and enforce the complementarity of x and $\overline{x}$. A literal has truth value *true* if it is equal to 0, and truth value *false* if it is equal to 1. Each clause C is translated to a monomial m with the property that $m = 0$ if and only if C is satisfied. For example, the clause $x \vee y \vee \overline{z}$ is translated to the monomial $xy\overline{z}$. There are two inference rules, *linear combination* $\frac{p \quad q}{\alpha p + \beta q}$ and *multiplication* $\frac{p}{xp}$, where p and q are (previously derived) polynomials, the coefficients α, β are elements of $\mathbb{F}$, and x is any variable (with or without a bar). These rules are sound in the sense that if the antecedent polynomials evaluate to zero under some assignment, then so does the consequent polynomial. A CNF formula F is refuted in PCR by deriving the constant term 1 from the (monomials corresponding to the) clauses in F.

The *size*, *degree*, and *monomial space* measures are analogs 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.



(a) Pyramid graph Π_2 of height 2.

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

(b) Pebbling contradiction Peb_{Π_2}.

Fig. 2. Pebbling contradiction Peb_{Π_2} for the pyramid graph Π_2 of height 2.

Let us now discuss why the method we use to prove Theorem 3.1 is unlikely to generalize to PCR. An example of formulas that seem hard to deal with in this way are so-called *pebbling contradictions*, which we briefly describe next.

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

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

(see Figure 2 for an illustration). Ben-Sasson [2009] showed that pebbling contradictions exhibit space-width trade-offs in resolution in that they can always be refuted in constant width as well as in constant space but that there are graphs for which optimizing one of these measures necessarily causes essentially worst-case linear behavior 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 unit clause v using the pebbling axiom for v and the unit clauses for its predecessors. When the refutation reaches z it derives a contradiction with the sink axiom $\overline{z}$ (see Figure 3(a) for an example). This refutation can always be carried out in constant width but for some graphs requires large space.

The other approach is a “top-down” refutation due to Ben-Sasson [2009] where one starts with the sink axiom $\overline{z}$ and derives clauses of the form $\overline{v}_1 \vee \cdots \vee \overline{v}_\ell$. A new clause is derived by replacing any vertex v_i in the old one by all its predecessors, that is, by resolving with the pebbling axiom for v_i. Since G is acyclic we can repeat this process until we get to the sources, for which the negated literals can be resolved away using source axioms. This refutation is illustrated in Figure 3(b). It is not hard to see that it can be performed in constant clause space, but it might require large width.

A careful study now reveals that the transformation of configurations in our proof of Theorem 3.1 maps either of the two refutations described previously into the other one. Instead of providing a formal argument, we encourage the reader to compute the transformations of the refutations in Figure 3(a)and 3(b), observing that the axioms are downloaded in opposite order in the two derivations. This observation is the main reason why our proof does not seem to generalize to PCR, as we now explain.

In PCR, we can represent any conjunction of literals $a_1 \wedge \cdots \wedge a_r$ as the binomial $1 - \prod_i \overline{a}_i$. Using this encoding with the bottom-up approach yields a third refutation,

1.	u	Axiom
2.	v	Axiom
3.	w	Axiom
4.	$\overline{u} \lor \overline{v} \lor x$	Axiom
5.	$\overline{v} \lor x$	$\mathrm{Res}(1, 4)$
6.	x	$\mathrm{Res}(2, 5)$
7.	$\overline{v} \lor \overline{w} \lor y$	Axiom
8.	$\overline{w} \lor y$	$\mathrm{Res}(2, 7)$
9.	y	$\mathrm{Res}(3, 8)$
10.	$\overline{x} \lor \overline{y} \lor z$	Axiom
11.	$\overline{y} \lor z$	$\mathrm{Res}(6, 10)$
12.	z	$\mathrm{Res}(9, 11)$
13.	$\overline{z}$	Axiom
14.	$\bot$	$\mathrm{Res}(12, 13)$

(a) Bottom-up refutation of Peb_{Π_2}.

1.	$\overline{z}$	Axiom
2.	$\overline{x} \lor \overline{y} \lor z$	Axiom
3.	$\overline{x} \lor \overline{y}$	$\mathrm{Res}(1, 2)$
4.	$\overline{v} \lor \overline{w} \lor y$	Axiom
5.	$\overline{v} \lor \overline{w} \lor \overline{x}$	$\mathrm{Res}(3, 4)$
6.	$\overline{u} \lor \overline{v} \lor x$	Axiom
7.	$\overline{u} \lor \overline{v} \lor \overline{w}$	$\mathrm{Res}(5, 6)$
8.	w	Axiom
9.	$\overline{u} \lor \overline{v}$	$\mathrm{Res}(7, 8)$
10.	v	Axiom
11.	$\overline{u}$	$\mathrm{Res}(9, 10)$
12.	u	Axiom
13.	$\bot$	$\mathrm{Res}(11, 12)$

(b) Top-down refutation of Peb_{Π_2}.

Fig. 3. Example resolution refutations of pebbling contradiction Peb_{Π_2}.

which has constant space but possibly large degree: the fact that a set of vertices U "are true" can be stored as the high-degree binomial $1 - \prod_{v \in U} \overline{v}$ instead of as a collection of low-degree monomials $\{v \mid v \in U\}$. Hence, there are constant space PCR refutations of pebbling contradictions in both the bottom-up and the top-down directions. This in turn means that if our proof method were to work for PCR, we would need to find constant degree refutations in both directions. For the top-down case it seems unlikely that such a refutation exists, since clauses of the form $\bigvee_{v \in U} \overline{v}$ cannot be represented as low-degree polynomials.

6. CONCLUDING REMARKS

In this work, we present an alternative, more explicit, proof of the result by Atserias and Dalmau [2008] 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 [2001] and showing that when a refutation has made medium progress toward 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 Atserias and Dalmau [2008] it is also refutable in constant width, and a simple counting argument then shows that exhaustive search in small width will find a polynomial-length resolution refutation. But is there any way of obtaining such a short refutation from a refutation in small space that is more explicit than doing exhaustive search? And can we obtain a short refutation without blowing up the space by more than, say, a constant factor? Known length-space trade-off results for resolution in Beame et al. [2012], Ben-Sasson and Nordström [2011], Beck et al. [2013], and Nordström [2009] 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 [Atserias et al. 2014] shows that there are formulas refutable in space and width s where the shortest refutation has length $n^{\Omega(s)}$, that is, matching the exhaustive search upper bound up to a (small) constant factor in the exponent.

An even more intriguing question is how the space and degree measures are related in polynomial calculus, as discussed in Section 5. Most relations between length, space, and width in resolution carry over with little or no modification to size, space, and degree, respectively, in polynomial calculus. So can it be that space also yields an upper bound on degree in polynomial calculus? Or could perhaps even the stronger claim hold that polynomial calculus space is an upper bound on resolution width? These questions remain wide open, but in the recent paper by Filmus et al. [2013] 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 Filmus et al. [2013] for a more detailed discussion of these issues.

ACKNOWLEDGMENTS

The authors wish to thank Albert Atserias, Ilario Bonacina, Nicola Galesi, and Li-Yang Tan for stimulating discussions on topics related to this work. We would also like to thank Alexander Razborov for sharing his proof of the theorem that space provides an upper bound on width, which is very similar to ours, although expressed in a different language. Finally, we are grateful to the anonymous reviewers, who helped improve this article considerably.

REFERENCES

Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. 2002. Space complexity in propositional calculus. *SIAM J. Comput.* 31, 4 (2002), 1184–1211. Preliminary version appeared in *STOC’00*.

Albert Atserias and Víctor Dalmau. 2008. A combinatorial characterization of resolution width. *J. Comput. System Sci.* 74, 3 (May 2008), 323–334. Preliminary version appeared in *CCC’03*.

Albert Atserias, Massimo Lauria, and Jakob Nordström. 2014. Narrow proofs may be maximally long. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC’14)*. 286–297.

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

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

Eli Ben-Sasson. 2009. Size space tradeoffs for resolution. *SIAM J. Comput.* 38, 6 (May 2009), 2511–2525. Preliminary version appeared in *STOC’02*.

Eli Ben-Sasson and Nicola Galesi. 2003. Space complexity of random formulae in resolution. *Random Structures and Algorithms* 23, 1 (Aug. 2003), 92–109. Preliminary version appeared in *CCC’01*.

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

Eli Ben-Sasson and Jakob Nordström. 2011. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS’11)*. 401–416. Full-length version available at http://eccc.hpi-web.de/report/2010/125/.

Eli Ben-Sasson and Avi Wigderson. 2001. Short proofs are narrow-Resolution made simple. *J. ACM* 48, 2 (March 2001), 149–169. Preliminary version appeared in *STOC’99*.

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

Vašek Chvátal and Endre Szemerédi. 1988. Many hard examples for resolution. *J. ACM* 35, 4 (Oct. 1988), 759–768.

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

Juan Luis Esteban and Jacobo Torán. 2001. Space bounds for resolution. *Information and Computation* 171, 1 (2001), 84–97. Preliminary versions of these results appeared in *STACS'99* and *CSL'99*.

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

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

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

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

Jakob Nordström. 2009. A simplified way of proving trade-off results for resolution. *Inform. Process. Lett.* 109, 18 (Aug. 2009), 1030–1035. Preliminary version appeared in ECCC report TR07-114, 2007.

Jakob Nordström. 2013. Pebble games, proof complexity and time-space trade-offs. *Log. Meth. Comput. Sci.* 9, 3, Article 15 (Sept. 2013), 15:1–15:63.

Alexander Razborov. 2014. Personal communication.

Alasdair Urquhart. 1987. Hard examples for resolution. *J. ACM* 34, 1 (Jan. 1987), 209–219.

Received June 2014; revised March 2015; accepted March 2015