Understanding Space in Proof Complexity: Separations and Trade-offs via Substitutions (Extended Abstract)

Eli Ben-Sasson[1] Jakob Nordström[2]

[1]Computer Science Department, Technion — Israel Institute of Technology, Haifa, 32000, Israel

[2]School of Computer Science and Communication, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden

eli@cs.technion.ac.il jakobn@kth.se

Abstract: For current state-of-the-art satisfiability algorithms based on the DPLL procedure and clause learning, the two main bottlenecks are the amounts of time and memory used. In the field of proof complexity, these resources correspond to the length and space of resolution proofs for formulas in conjunctive normal form (CNF). There has been a long line of research investigating these proof complexity measures, but while strong results have been established for length, our understanding of space and how it relates to length has remained quite poor. In particular, the question whether resolution proofs can be optimized for length and space simultaneously, or whether there are trade-offs between these two measures, has remained essentially open apart from a few results in restricted settings.

In this paper, we remedy this situation by proving a host of length-space trade-off results for resolution in a completely general setting. Our collection of trade-offs cover almost the whole range of values for the space complexity of formulas, and most of the trade-offs are superpolynomial or even exponential and essentially tight. Using similar techniques, we show that these trade-offs in fact extend (albeit with worse parameters) to the exponentially stronger k-DNF resolution proof systems, which operate with formulas in disjunctive normal form with terms of bounded arity k. We also answer the open question whether the k-DNF resolution systems form a strict hierarchy with respect to space in the affirmative.

Our key technical contribution is the following, somewhat surprising, theorem: Any CNF formula F can be transformed by simple variable substitution into a new formula F' such that if F has the right properties, F' can be proven in essentially the same length as F, whereas on the other hand the minimal number of *lines* one needs to keep in memory simultaneously in any proof of F' is lower-bounded by the minimal number of *variables* needed simultaneously in any proof of F. Applying this theorem to so-called pebbling formulas defined in terms of pebble games on directed acyclic graphs, we obtain our results.

Keywords: proof complexity; resolution; k-DNF resolution; length; space; separation; trade-off; pebble game; pebbling contradiction

1 Introduction

A central theme in the field of propositional proof complexity is the study of limitations of natural proof systems. Such a study is typically conducted by considering a *complexity measure* of propositional proofs and investigating under which circumstances this measure is large. The most thoroughly examined complexity measure is that of *proof size/length*. The interest in this measure is motivated by its connections to the P vs. NP problem (since by [22], proving superpolynomial lower bounds for arbitrary proof systems would separate NP and $\mathsf{co\text{-}NP}$, and hence show $\mathsf{P} \neq \mathsf{NP}$), by methods for proving independence results in first order theories of bounded arithmetic (for an example, see [2]), and because lower bounds on proof length imply lower bounds on the running time of algorithms for solving NP-complete problems such as SATISFIABILITY (such algorithms are usually referred to as *SAT solvers*).

Proof Space

This paper focuses on a more recently suggested complexity measure known as *space*. This measure was first defined and studied by Esteban and Torán [25] in the context of the famous *resolution* proof system [18], which is a proof system for refuting unsatisfiable formulas in conjunctive normal form, henceforth *CNF formulas*. The space measure was subsequently generalized to other proof systems by Alekhnovich et al. in [4]. Roughly speaking, the space of a proof corresponds to the minimal size of a blackboard needed to give a self-contained presentation of the proof, where the correctness of each step is verifiable from what is currently on the blackboard. The interest in space complexity stems from two main sources that we survey next.

First, there are intricate and often surprising connections between the space and length complexity measures. For resolution, it follows from the elegant results of Atserias and Dalmau [6] that upper bounds on space imply upper bounds on length. Esteban and Torán [25] showed the converse that length upper bounds imply space upper bounds for the restricted case of *tree-like resolution*. Recall that the *tree-like* version of a *sequential*[1] proof system has the added constraint that every line in the proof can be used at most once to derive a subsequent line. In terms of space, a proof is tree-like if any claim appearing on the blackboard must be erased immediately after it has been used to derive a new claim.

Another related question which has attracted interest is whether space and length can display *trade-offs*, that is whether there are formulas having proofs in both short length and small space, but for which there are no proofs in short length and small space *simultaneously*. Such length-space trade-offs have been established in restricted settings by the current authors in [11, 37][2] but nothing has been known for refutations of explicit formulas in general, unrestricted resolution.

A second motivation to study space is because of its connection to the memory consumption of SAT solvers. For instance, the family of backtracking heuristics suggested by [23, 24] and known as *Davis-Putnam-Logemann-Loveland (DPLL)* SAT solvers have the following property. When given as input an unsatisfiable CNF formula F, the description of the execution of a DPLL SAT solver corresponds to a tree-like resolution proof refuting F. Thus, lower bounds on tree-like refutation space imply lower bounds on the *memory consumption* of DPLL SAT solvers, much like lower bounds on tree-like refutation length imply lower bounds on the *running time* of DPLL heuristics.

During the last 10-15 years, a family of SAT solvers known as *DPLL with clause learning* [9, 35] (which we denote by DPLL+) has been put to practical use with impressive success. For instance, an overwhelming majority of the best algorithms in recent rounds of the international SAT competitions [42] belong to this class. These SAT solvers have the property that an execution trace corresponds to a (non-tree-like) resolution refutation. Hence, space lower bounds in general resolution can be thought of as corresponding to memory lower bounds for these algorithms, and length-space trade-offs could have implications for trade-offs between time efficiency and memory consumption.

We end this discussion by pointing out that there is still much left to explore regarding the connection between space lower bounds in proof complexity and memory consumption of SAT solvers. On the one hand, the memory consumption of a "typical" DPLL+ SAT solver can be substantially larger than the theoretical *upper* lower bounds that are guaranteed to hold for the space complexity of any CNF formula, so in this sense any *lower* bounds on refutation space will suffer from the inherent limitation that they seem "too small." On the other hand, the theoretical lower bounds on refutation space are *worst-case* bounds for *non-deterministic algorithms*, and hence apply even to the most memory-efficient proofs theoretically possible. This seems to be a very different scenario from the kind of proofs produced by a typical SAT solver, which has no way of "magically"

[1] A proof system is said to be *sequential* if a proof π in the system is a *sequence* of lines $\pi = \{L_1, \ldots, L_\tau\}$ where each line is derived from previous lines by one of a finite set of allowed *inference rules*.

[2] A related result, claimed in [29], has later been retracted by the authors in [30].

knowing exactly which clauses to keep in memory at which point in the proof. Understanding what kind of practical implications one can get on the memory consumption of SAT solvers from refutation space lower bounds thus remains an interesting open problem.

k-DNF Resolution

The family of sequential proof systems known as *k-DNF resolution* was introduced by Krajíček [32] as a intermediate step between resolution and depth-2 Frege. Roughly speaking, for integers $k > 0$ the kth member of this family, denoted henceforth by $\mathfrak{R}(k)$, is only allowed to reason in terms of formulas in disjunctive normal form (*DNF formulas*) with the added restriction that any conjunction in any formula is over at most k literals. For $k = 1$, the lines in the proof are hence disjunctions of literals, and the proof system $\mathfrak{R}(1) = \mathfrak{R}$ is simply standard resolution. At the other extreme, $\mathfrak{R}(\infty)$ is equivalent to depth-2 Frege.

The original motivation to study this family of proof systems, as stated in [32], was to better understand the complexity of counting in weak models of bounded arithmetic, and it was later observed that these systems are also related to SAT solvers that reason using multi-valued logic (see [31] for a discussion of this point). By now a number of works have shown superpolynomial lower bounds on the length of $\mathfrak{R}(k)$-refutations, most notably for (various formulations of) the pigeonhole principle and for random CNF formulas [3, 5, 7, 31, 41, 43, 44]. Of particular relevance to our current work are the results of Segerlind et al. [44] and of Segerlind [43] showing that the family of $\mathfrak{R}(k)$-systems form a *strict hierarchy* with respect to proof length. More precisely, in these papers it is shown that for every integer $k > 0$ one can find a family of formulas $\{F_n\}_{n=1}^{\infty}$ of arbitrarily large size n such that F_n has a $\mathfrak{R}(k+1)$-refutation of polynomial length $n^{O(1)}$ but all $\mathfrak{R}(k)$-refutations of F_n require exponential length $2^{n^{\epsilon}}$ for some constant $\epsilon > 0$.

Just as in the case for standard resolution, the understanding of space complexity in k-DNF resolution has remained more limited. We are aware of only one prior work by Esteban et al. [27] shedding light on this question. Their paper establishes essentially optimal linear space lower bounds for $\mathfrak{R}(k)$ and also prove that the family of *tree-like* $\mathfrak{R}(k)$ systems form a strict hierarchy with respect to space. What they show is that there exist arbitrarily large formulas F_n of size n that can be refuted in tree-like $\mathfrak{R}(k+1)$ in constant space but require space $\Omega(n/\log^2 n)$ to be refuted in tree-like $\mathfrak{R}(k)$.

It should be pointed out, however, that as observed in [27, 32] the family of tree-like $\mathfrak{R}(k)$ systems for all $k > 0$ are strictly weaker than standard resolution. As was noted above, the family of general, unrestricted $\mathfrak{R}(k)$ systems are strictly stronger than resolution, so the results in [27] leave completely open the question of whether there is a strict space hierarchy for (non-tree-like) $\mathfrak{R}(k)$ or not.

(Informal) Definition of Length and Space

As a final point before turning to our results, we briefly and informally recall what is meant by "length" and "space." Following [4], we view a refutation of an unsatisfiable CNF formula F as being presented on a blackboard. The refutation is represented as a sequence of sets of k-DNF formulas $\pi = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$, where $\mathbb{D}_t$ is a snapshot of the blackboard at time t in the refutation. In particular, $\mathbb{D}_0$ should be the empty set, $\mathbb{D}_\tau$ should contain the contradictory empty formula, and at time t we can go from $\mathbb{D}_{t-1}$ to $\mathbb{D}_t$ by *(i)* writing a clause of F (an *axiom*) on the blackboard, *(ii)* erasing a line from the board, or *(iii)* inferring a new line from those lines present on the board according to the inference rules of k-DNF resolution. We do not discuss the details of these rules here, since the exact definitions in fact do not matter—the lower bounds we prove hold for any *arbitrarily strong* (but sound) rules. What is important is that the only new formulas that can be derived at any given point in time are those implied by the set of formulas that are currently on the blackboard, and that these formulas are all k-DNFs.

The length of a refutation is the number of formulas appearing in the refutation counted with repetitions, or equivalently (within a factor of 2) the number of derivation steps. There are several different ways to measure the space of a set $\mathbb{D}_t$ in our refutation. The crudest way is to count the number of k-DNF formulas on the board, i.e., to measure the size of $\mathbb{D}_t$. We call this the *formula space*, or simply, *space* of $\mathbb{D}_t$. (For resolution, i.e., when

$k = 1$, this is the well-studied measure of *clause space*.) A finer granulation is to measure the *total space*—the number of appearances of literals in $\mathbb{D}_t$, counted with repetitions. Formula space and total space are the two space measures that have received the most attention in previous research, and they are also the focus of the current paper. A third, closely related, measure that will also be of interest to us is *variable space*, defined as the number of distinct variables appearing on the board. It is easily seen that variable space is a lower bound on total space. For all of these space measures, the space of a refutation $\pi = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ is the maximal space of any k-DNF set $\mathbb{D}_t$ in it.

1.1 Our Results in Brief

Space Hierarchy for k-DNF Resolution

Our first main result is that Krajíček's family of k-DNF resolution proof systems form a strict hierarchy with respect to space. More precisely, we separate k-DNF resolution from $(k+1)$-DNF resolution in the sense that we exhibit for every k a family of *explicitly constructible*[3] CNF formulas of size n that can be refuted in constant formula space and linear length simultaneously in $(k+1)$-DNF resolution (i.e., they are very easy with respect to both measures) but have the property that any k-DNF resolution, no matter how long or short, must by necessity use at least order of $\sqrt[k+1]{n}$ formula space.

Length-Space Trade-offs

Our second main result is a collection of strong length-space trade-offs for k-DNF resolution. For $k = 1$, i.e., standard resolution, these are the first trade-off results for resolution refutations of explicit formulas in the general, unrestricted resolution proof system, thus eliminating all technical restrictions in the previous works [11, 37]. For $k > 1$, to the best of our knowledge no trade-offs have been known even in restricted settings.

We also want to emphasize two other novel aspects of our results. First, as was discussed above there are several different ways of measuring space, and previous papers have focused on one particular measure and proven results specifically for that measure. Our techniques, however, allow us to obtain results that hold for both total space and formula space (i.e., the largest and smallest space measure) simultaneously. Second, our upper bounds hold for the standard *syntactic* version of the proof systems, where new formulas can be derived from two existing formulas by a limited set of structural rules, whereas the lower bounds hold for *semantic* versions where new formulas can be derived from an unlimited set of formulas by arbitrary sound rules. The reason this is worth pointing out is that in general, semantic k-DNF resolution proof systems are known to be exponentially stronger than syntactic systems.

We give the formal statements of our trade-offs in Section 2, but a general template for the kind of trade-off theorems we are able to prove is as follows.

Theorem 1 (Trade-off theorem template (informal)). *Let K be a fixed positive integer and let $s_{\mathrm{lo}}(n)$ and $s_{\mathrm{hi}}(n)$ be suitable functions such that $s_{\mathrm{lo}}(n) \ll s_{\mathrm{hi}}(n) = \mathrm{O}(n/\log\log n)$. Then there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ and width $\mathrm{O}(1)$ (with constants depending on K) such that the following holds:*

- *The formulas F_n are refutable in syntactic resolution in (small) total space $\mathrm{O}(s_{\mathrm{lo}}(n))$.*
- *There are also syntactic resolution refutations π_n of F_n in simultaneous length $\mathrm{O}(n)$ and (much larger) total space $\mathrm{O}(s_{\mathrm{hi}}(n))$.*
- *However, any resolution refutation, even semantic, in formula space $\mathrm{o}(s_{\mathrm{hi}}(n))$ must have superpolynomial or sometimes even exponential length.*
- *Even for the much stronger semantic k-DNF resolution proof systems, $k \leq K$, it holds that any $\mathfrak{R}(k)$-refutation of F_n in formula space $\mathrm{o}\big(\sqrt[k+1]{s_{\mathrm{hi}}(n)}\big)$ must have superpolynomial length (or exponential length, correspondingly).*

We instantiate this theorem template for a wide range of space functions $s_{\mathrm{lo}}(n)$ and $s_{\mathrm{hi}}(n)$ from constant space all the way up to nearly linear space. This is in contrast to [37], where the trade-off results are obtained only for a very carefully selected ratio of space to formula size. Moreover, our trade-offs are robust in that they are not sensitive to small variations in either length or space (as in [37]). That is, intuitively speaking they will not

[3] A family of formulas is *explicitly constructible* if there exists a polynomial time algorithm that on input 1^n produces the nth member of the family.

show up only for a SAT solver being unlucky and picking just the wrong threshold when trying to hold down the memory consumption. Instead, any resolution refutation having length or space in the same general vicinity will be subject to the same qualitative trade-off behaviour.

1.2 Overview of Technical Contributions

We want to highlight three technical contributions underpinning the results discussed informally above.

Substitution Space Theorems

Our first key technical contribution is a general way to derive strong space lower bounds in resolution from weak lower bounds on the number of variables that occur simultaneously in a proof. Very loosely, we show the following: Suppose we have a formula that has short refutations but where any such short refutation must mention many variables at some point. Then by making variable substitutions in this formula and expanding the result into a CNF formula in the natural way, this new formula will still have short refutations, but now any such refutation must use lots of space, in the sense that lower bounds on *variable space* will translate into lower bounds on *formula space*.

We believe that this generic procedure of transforming weak space lower bounds into stronger ones is an interesting result in its own right that sheds new light on space measures in proof complexity. To support this point, we strengthen the theorem by showing that not only can we obtain strong resolution lower bounds from weak lower bounds in resolution in this way, but it is also possible to lift weak resolution lower bounds to strong lower bounds in other more powerful proof systems, namely k-DNF resolution systems. We remark that this general idea of "hardness amplification" in proof complexity has also been used in the recent work of Beame et al. [10], although the actual techniques there appear somewhat orthogonal to those in the current paper (and, in particular, incomparable in the sense that it seems neither paper can be used to derive the results in the other).

Minimally Unsatisfiable k-DNF Sets

One crucial ingredient in the proof of the substitution space theorem for resolution is analyzing the structure of sets of disjunctive clauses that imply many other clauses. Intuitively, it seems reasonable that if the set of implied clauses is sufficiently large and disjoint, the set of clauses implying all these clauses cannot itself be too small. One important special case of this is for clause sets containing many variables but being *minimally unsatifiable*—that is, every clause places a necessary constraint on the variables to enforce unsatifiability and if just one arbitrary clause is removed from the set, then the rest can be satisfied. It is well known that such a clause set must contain strictly more clauses than variables, and we can use similar proof techniques to derive the more general result that we need.

When we want to extend our theorem to k-DNF resolution, it becomes essential to understand instead the structure of sets of k-DNF formulas that imply many other k-DNF formulas. Here there are no previous results to build on, as the proof techniques that yield tight results for disjunctive clauses can be shown to break down fundamentally. Instead, we have to develop new methods. One important step along the way is to understand the structure of minimally unsatisfiable sets of k-DNF formulas, which appears to be a natural combinatorial problem of independent interest. We prove that a minimally unsatisfiable k-DNF set of size m can contain at most $\lessapprox m^{k+1}$ variables, and this bound turns out to be tight up to an additive one in the exponent in view of recent joint work [40] of Razborov and the second author.

Reductions Between Resolution and Pebbling

Using the substitution space theorems, we can construct reductions between (k-DNF) resolution on the one hand and so-called pebble games played on directed acyclic graphs (DAGs) on the other. In one direction, this reduction is easy, but the other direction is nontrivial. Moreover, our reductions are time- and space-preserving. This allows us (modulo some technical complications which we ignore for the moment) to translate known trade-off results for pebbling into corresponding trade-offs for resolution. This is done by transforming the pebble game played on a DAG G into a CNF formula that encodes this particular problem instance of the game, and showing that this formula has similar trade-off properties in resolution as the DAG G has for the pebble game.

With hindsight, such a correspondence might seem more or less obvious, so let us stress that this is not the case. Pebble games on graphs and $\mathfrak{R}(k)$-refutations of CNF formulas are very different objects. Once we have translated a pebbling instance into a CNF formula, it is not at all clear why an $\mathfrak{R}(k)$-prover refuting this formula would have to care about how it was constructed. There might be shortcuts in the proof complexity world that do not correspond to anything meaningful in the pebbling world. And indeed, reading previous literature on pebbling formulas in proof complexity reveals a few such surprising shortcuts, and there has been no consensus on what properties these formulas are likely to have in general.

What we show is that for the right flavour of pebbling formulas, any prover refuting such formulas must in effect reason in terms of pebblings. More precisely, we show that given any $\mathfrak{R}$(k)-refutation, no matter how it is structured, we can extract from it a pebbling of the underlying DAG, and this pebbling has at least as good time and space properties as the refutation from which it was extracted. In other words, the pebbling formula inherits the time-space trade-off properties of the DAG in terms of which it is defined. This allows us to draw on the rich literature on pebbling trade-offs from the 1970s and 1980s, as well as on newer results by the second author in [38], to obtain strong trade-offs in proof complexity.

1.3 Organization of the Rest of This Paper

In Section 2, we present formal statements of our main results. Then, in order not to let all the notation and terminology obscure what is in essence a clean and simple proof construction, in Section 3 we briefly outline some of the key ingredients in the proofs. We refer the reader to the full-length version [15] for the details. Concluding this extended abstract, in Section 4 we briefly discuss some open questions.

2 Formal Statements of Results

In what follows, let us write $L(\pi)$ to denote the length of a resolution refutation and $Sp(\pi)$, $TotSp(\pi)$, and $VarSp(\pi)$ to denote the formula space, total space and variable space, respectively. Taking the minimum over all resolution refutations of F, we let $L_{\mathfrak{R}}(F \vdash 0)$ denote the length of a shortest refutation, and $Sp_{\mathfrak{R}}(F \vdash 0)$, $TotSp_{\mathfrak{R}}(F \vdash 0)$, and $VarSp_{\mathfrak{R}}(F \vdash 0)$ are defined completely analogously. These definitions are also generalized to $\mathfrak{R}(k)$ for general k. To state our results it will also convenient to use the notation $W(\pi)$ for the *width* of a standard resolution refutation, i.e., the size of a largest clause in it, and $W_{\mathfrak{R}}(F \vdash 0)$ for the minimal width of any standard resolution refutation of F. See [15] for more formal definitions of these concepts.

2.1 Substitution Space Theorems

If F is a CNF formula over variables $x, y, z, \ldots$ and $f : \{0,1\}^d \mapsto \{0,1\}$ is a Boolean function over d variables, we can obtain a new CNF formula by substituting $f(x_1, \ldots, x_d)$ for every variable x and then expand to conjunctive normal form. We will write $F[f]$ to denote the resulting *substitution formula*. For example, for the disjunctive clause $C = x \vee \overline{y}$ and the exclusive or function $f = x_1 \oplus x_2$ we have that

$$\begin{aligned} C[f] = \quad & (x_1 \vee x_2 \vee y_1 \vee \overline{y}_2) \\ & \wedge (x_1 \vee x_2 \vee \overline{y}_1 \vee y_2) \\ & \wedge (\overline{x}_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2) \\ & \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2) \ . \end{aligned} \tag{1}$$

We say that f is *k-non-authoritarian* if no partial assignment to any subset of k variables can fix the value of f to true or false and that f is *k-authoritarian* otherwise. For instance, the XOR function $\oplus$ on $d+1$ variables is d-non-authoritarian, as is the majority function on $2d+1$ variables. If f is 1-non-authoritarian (1-authoritarian) we say that the function is simply *non-authoritarian* (*authoritarian*). For example, non-exclusive or $\vee$ of any arity is always authoritarian.

Loosely put, the substitution space theorem for resolution says that if a CNF formula F can be refuted in resolution in small length and width simultaneously, then so can the substitution formula $F[f_d]$. There is an analogous result in the other direction as well in the sense that we can translate any refutation π_f of $F[f_d]$ into a refutation π of the original formula F where the length of π is almost upper-bounded by the length of π_f (this will be made precise below). So far this is nothing very unexpected, but what is more interesting is that if f_d is non-authoritarian, then the clause space of π_f is an upper bound on the number of variables mentioned simultaneously in π.

Thus, the theorem says that we can convert (weak) lower bounds on variable space into (strong) lower bounds on clause space by making substitutions using non-authoritarian functions.

Theorem 2 (Substitution space theorem for resolution). *Let F be any unsatisfiable CNF formula and f_d be any non-constant Boolean function of arity d. Then it holds that the substitution formula $F[f_d]$ can be refuted in resolution in width*

$$W_{\mathfrak{R}}\bigl(F[f_d] \vdash 0\bigr) = \mathrm{O}\bigl(d \cdot W_{\mathfrak{R}}(F \vdash 0)\bigr) \ ,$$

length

$$L_{\mathfrak{R}}\bigl(F[f_d] \vdash 0\bigr) \leq \min_{\pi : F \vdash 0}\bigl\{L(\pi) \cdot \exp\bigl(\mathrm{O}(d \cdot W(\pi))\bigr)\bigr\} \ ,$$

and total space

$$\mathit{TotSp}_{\mathfrak{R}}\bigl(F[f_d] \vdash 0\bigr) \leq \min_{\pi : F \vdash 0}\bigl\{\mathit{TotSp}(\pi) \cdot \exp\bigl(\mathrm{O}(d \cdot W(\pi))\bigr)\bigr\} \ .$$

In the other direction, any semantic resolution refutation $\pi_f : F[f_d] \vdash 0$ of the substitution formula can be transformed into a syntactic resolution refutation $\pi : F \vdash 0$ of the original formula such that the number of axiom downloads[4] in π is at most the number of axiom downloads in π_f. If in addition f_d is non-authoritarian, it holds that $Sp(\pi_f) > VarSp(\pi)$, i.e., the clause space of refuting the substitution formula $F[f_d]$ is lower-bounded by the variable space of refuting the original formula F.

Note that if F is refutable simultaneously in linear length and constant width, then the bound in Theorem 2 on $L\bigl(F[f_d] \vdash 0\bigr)$ becomes linear in $L(F \vdash 0)$.

The substitution space theorem for k-DNF resolution extends Theorem 2 by telling us that for k-non-authoritarian functions f, we can translate back and forth between standard resolution refutations of F and $\mathfrak{R}(k)$-refutations of the substitution formula $F[f]$ in a (reasonably) length- and space-preserving way. When the "proof blackboard" contains k-DNFs instead of disjunctive clauses, the analysis becomes much more challenging, however, and the bounds we are able to obtain become correspondingly worse. Below, we state the theorem with asymptotic factors hidden by the asymptotic notation to make it easier to parse. The complete version is given in [15].

Theorem 3 (Substitution space theorem for k-DNF resolution). *Let F be any unsatisfiable c-CNF formula and f_d be any non-constant Boolean function of arity d, and suppose furthermore that c, d, and k are universal constants. Then the following two properties hold for the substitution formula $F[f_d]$:*

1. *If F can be refuted in syntactic standard resolution in length L and total space s simultaneously, then $F[f_d]$ can be refuted in syntactic $\mathfrak{R}(d)$ in length $\mathrm{O}(L)$ and total space $\mathrm{O}(s)$ simultaneously.*
2. *If f_d is k-non-authoritarian and $F[f_d]$ can be refuted by a semantic $\mathfrak{R}(k)$-refutation that requires formula space s and makes L axiom downloads, then F can be refuted by a syntactic standard resolution refutation that requires variable space at most $\mathrm{O}\bigl(s^{k+1}\bigr)$ and makes at most L axiom downloads.*

The proofs of Theorems 2 and 3 are inspired by our recent work [13] and indeed our main theorem there can be seen to follow from Theorem 2. Let us discuss the new aspects of the more general theorems presented in this paper. First and foremost, our results extend to $\mathfrak{R}(k)$ for $k > 1$ whereas the previous theorem applies only to resolution. Second, our previous statement only holds for a very special kind of formulas (namely the pebbling formulas discussed above) whereas Theorems 2 and 3 can be used to convert *any* CNF formula requiring large variable space into a new and closely related CNF formula requiring large formula space. Third, in this paper we get length-preserving as well as space-preserving reductions, whereas it was unclear how to derive similar reductions from our previous work. And length-

[4] It would have been nice if the bound in terms of number of axiom downloads could have be strengthened to a bound in terms of length, but this is *not* true. The reason for this is that the proof refuting $F[f_d]$ is allowed to use any arbitrarily strong *semantic* inference rules, and this can lead to exponential savings compared to syntactic resolution. To see this, just let F be an encoding of, say, the pigeonhole principle and let π_f be the refutation that downloads all axioms of $F[f_d]$ and then derives contradiction in one step. Luckily enough, though, the bound in terms of axiom downloads turns out to be exactly what we need for our applications.

preserving reductions are crucial for our length-space trade-offs described below.

We will return to these theorems and sketch the main ingredients in the proofs in Sections 3.1 and 3.2, but before that we want to describe why these tools will be so useful for us. We do so next.

2.2 Translating Refutations to Pebblings

The *pebble game* played on a DAG G models the calculation described by G, where the source vertices contain the inputs and non-source vertices specify operations on the values of the predecessors. Placing a pebble on a vertex v corresponds to storing in memory the partial result of the calculation described by the subgraph rooted at v. Removing a pebble from v corresponds to deleting the partial result of v from memory. Black pebbles correspond to deterministic computation and white pebbles to nondeterministic guesses. A *pebbling* $\mathcal{P}$ of G is a sequence of moves starting with the graph being completely empty and ending with all vertices empty except for a black pebble on the (unique) sink vertex. The *time* of a pebbling is the number of pebbling moves and the *space* is the maximal number of pebbles needed at any point during the pebbling.

The pebble game on the graph G can be encoded as an unsatisfiable CNF formula Peb_G saying that the sources of G are true and that truth propagates through the graph in accordance with the pebbling rules, but that the sink is false. Given any black-only pebbling $\mathcal{P}$ of G, we can mimic this pebbling in a resolution refutation of Peb_G by deriving that a literal v is true whenever the corresponding vertex in G is pebbled (this was perhaps first observed in [17]).

Lemma 4 ([17]). *Let G be a DAG with unique sink and bounded vertex indegree. Then given any complete black pebbling $\mathcal{P}$ of G, we can construct a standard resolution refutation $\pi : Peb_G \vdash 0$ such that $L(\pi) = \mathrm{O}(\mathsf{time}(\mathcal{P}))$, $W(\pi) = \mathrm{O}(1)$, and $TotSp(\pi) = \mathrm{O}(\mathsf{space}(\mathcal{P}))$.*

In the other direction, we start with the result of the first author [11] that if we take any refutation of a pebbling contradiction and let positive and negative literals correspond to black and white pebbles respectively, then we get (essentially) a legal black-white pebbling of the underlying DAG. That is not quite what we need, however, since it only provides a weak bound in terms of variable space.

This is where Theorems 2 and 3 come into play. If we make substitutions in Peb_G with suitably non-authoritarian functions, the upper bounds in Lemma 4 remain true (with adjustments in constant factors), while the lower bounds are lifted from variable space to formula space. For simplicity, we only state the lower bounds in the special case for standard resolution below.

Theorem 5. *Let f be any non-authoritarian Boolean function and G be any DAG with unique sink and bounded indegree. Then from any standard resolution refutation $\pi : Peb_G[f] \vdash 0$ we can extract a black-white pebbling strategy $\mathcal{P}_\pi$ for G such that $\mathsf{time}(\mathcal{P}_\pi) = \mathrm{O}(L(\pi))$ and $\mathsf{space}(\mathcal{P}_\pi) = \mathrm{O}(Sp(\pi))$.*

2.3 Space Separations and Length-Space Trade-offs

Combining the theorems in Section 2.1 with the reductions between resolution and pebble games in Section 2.2, we can now establish our space separation and length-space trade-off results. Let us start by formally stating the space hierarchy theorem for $\mathfrak{R}(k)$.

Theorem 6 (k-DNF resolution space hierarchy). *For every $k \geq 1$ there exists an explicitly constructible family $\{F_n\}_{n=1}^{\infty}$ of CNF formulas of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that*

- *there are $\mathfrak{R}(k+1)$-refutations $\pi_n : F_n \vdash 0$ in simultaneous length $L(\pi_n) = \mathrm{O}(n)$ and formula space $Sp(\pi_n) = \mathrm{O}(1)$, but*
- *any $\mathfrak{R}(k)$-refutation of F_n requires formula space $\Omega\big(\sqrt[k+1]{n/\log n}\big)$.*

The constants hidden by the asymptotic notation depend only on k.

The families $\{F_n\}_{n=1}^{\infty}$ are obtained by considering pebbling formulas defined in terms of the graphs in [28] requiring pebbling space $\Theta(n/\log n)$, and substituting a k-non-authoritarian Boolean function f of arity $k+1$, for instance XOR over $k+1$ variables, in these formulas.

Moving on to our length-space trade-offs, in the remainder of this section we try to highlight some of the results that we find to be the most interesting. A fuller and more detailed account of our collection of trade-off results is given in [15]. We reiterate that all of our results are for explicitly constructible formulas, and that in addition most of the constructions are actually very clean and transparent in that they are obtainable from pebbling formulas over simple families of DAGs.

From the point of view of space complexity, the easiest formulas are those refutable in constant total space, i.e., formulas having so simple a structure that there are resolution refutations where we never need to keep more than a constant number of symbols on the proof blackboard. A priori, it is not even clear whether we should expect that any trade-off phenomena could occur for such formulas, but it turns out that there are quadratic length-space trade-offs.

Theorem 7 (Quadratic trade-offs for constant space). *For any fixed positive integer K there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that the following holds (where all multiplicative constants hidden in the asymptotic notation depend only on K):*

- *The formulas F_n are refutable in syntactic resolution in total space $TotSp_{\mathfrak{R}}(F_n \vdash 0) = \mathrm{O}(1)$.*
- *For any $s_{\mathrm{hi}}(n) = \mathrm{O}\big(\sqrt{n}\big)$ there are syntactic resolution refutations π_n of F_n in simultaneous length $L(\pi_n) = \mathrm{O}\big((n/s_{\mathrm{hi}}(n))^2\big)$ and total space $TotSp(\pi_n) = \mathrm{O}(s_{\mathrm{hi}}(n))$.*
- *For any semantic resolution refutation $\pi_n : F_n \vdash 0$ in formula space (i.e., clause space) $Sp(\pi_n) \leq s_{\mathrm{hi}}(n)$ it holds that $L(\pi_n) = \Omega\big((n/s_{\mathrm{hi}}(n))^2\big)$.*
- *For any $k \leq K$, any semantic k-DNF resolution refutation π_n of F_n in formula space $Sp(\pi_n) \leq s_{\mathrm{hi}}(n)$ must have length $L(\pi_n) = \Omega\left(\left(n/\left(s_{\mathrm{hi}}(n)^{1/(k+1)}\right)\right)^2\right)$. In particular, any constant-space $\mathfrak{R}(k)$-refutation must also have quadratic length.*

Theorem 7 follows by combining our machinery with the seminal work on pebbling trade-offs by Lengauer and Tarjan [34] and the structural results on simulations of black-white pebblings by resolution by the second author in [38].

Remark 8. Notice that the trade-off applies to both formula space and total space. This is because the upper bound is stated in terms of the larger of these two measures (total space) while the lower bound is in terms of the smaller one (formula space). Note also that the upper bounds hold for the usual, syntactic versions of the proof systems, whereas the lower bounds hold for the much stronger semantic systems, and that for standard resolution the upper and lower bounds are tight up to constant factors. These properties of our results are inherited from the substitution space theorems, and they hold for all our trade-offs stated here. Finally, we remark that we have to pick some arbitrary but fixed limit K for the size of the terms when stating the results for k-DNF resolution, since for any family of formulas we consider there will be very length- and space-efficient $\mathfrak{R}(k)$-refutation refutations if we allow terms of unbounded size.

Our next result relies on a new pebbling trade-off result in [38], building on earlier work by Carlson and Savage [19, 20]. Using this new result, we can derive among other things the rather striking statement that for any *arbitrarily slowly growing* non-constant function, there are explicit formulas of such (arbitrarily small) space complexity that nevertheless exhibit *superpolynomial* length-space trade-offs.

Theorem 9 (Superpolynomial trade-offs for arbitrarily slowly growing space). *Let $s_{\mathrm{lo}}(n) = \omega(1)$ be any arbitrarily slowly growing function*[5] *and fix any $\epsilon > 0$ and positive integer K. Then there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that the following holds:*

- *The formulas F_n are refutable in syntactic resolution in total space $TotSp_{\mathfrak{R}}(F_n \vdash 0) = \mathrm{O}(s_{\mathrm{lo}}(n))$.*
- *There are syntactic resolution refutations π_n of the formulas F_n in simultaneous length $L(\pi_n) = \mathrm{O}(n)$ and total space $TotSp(\pi_n) = \mathrm{O}\left(\left(n/s_{\mathrm{lo}}(n)^2\right)^{1/3}\right)$.*
- *Any semantic resolution refutation of F_n in clause space $\mathrm{O}\left(\left(n/s_{\mathrm{lo}}(n)^2\right)^{1/3-\epsilon}\right)$ must have superpolynomial length.*
- *For any $k \leq K$, any semantic k-DNF resolution refutation of F_n in formula space*

[5]For technical reasons, let us also assume here that $s_{\mathrm{lo}}(n) = \mathrm{O}\big(n^{1/7}\big)$, i.e., that $s_{\mathrm{lo}}(n)$ does not grow too quickly. This restriction is inconsequential since for such fast-growing $s_{\mathrm{lo}}(n)$ other trade-off results presented below will yield much stronger bounds.

$\mathrm{O}\left(\left(n/s_{\mathrm{lo}}(n)^2\right)^{1/(3(k+1))-\epsilon}\right)$ *must have superpolynomial length.*

All multiplicative constants hidden in the asymptotic notation depend only on K, ϵ and s_{lo}.

Observe the robust nature of this trade-off, which is displayed by the long range of space complexity in standard resolution, from $\omega(1)$ up to $\approx n^{1/3}$, which requires superpolynomial length. Note also that the trade-off result for standard resolution is very nearly tight in the sense that the superpolynomial lower bound on length in terms of space reaches up to very close to where the linear upper bound kicks in.

The two theorems above focus on trade-offs for formulas of low space complexity, and the lower bounds on length obtained in the trade-offs are somewhat weak—the superpolynomial growth in Theorem 9 is something like $n^{s_{\mathrm{lo}}(n)}$. We next present a theorem that has both a stronger superpolynomial length lower bounds than Theorem 9 and an even more robust trade-off covering a wider (although non-overlapping) space interval. This theorem again follows by applying our tools to the pebbling trade-offs in [34].

Theorem 10 (Robust superpolynomial trade-off for medium-range space). *For any positive integer K, there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that the following holds (where the hidden constants depend only on K):*

- *The formulas F_n are refutable in syntactic resolution in total space $TotSp_{\mathfrak{R}}(F_n \vdash 0) = \mathrm{O}(\log^2 n)$.*
- *There are syntactic resolution refutations of F_n in length $\mathrm{O}(n)$ and total space $\mathrm{O}(n/\log n)$.*
- *Any semantic resolution refutation of F_n in clause space $Sp(\pi_n) = \mathrm{o}(n/\log n)$ must have length $L(\pi_n) = n^{\Omega(\log\log n)}$.*
- *For any $k \leq K$, any semantic $\mathfrak{R}(k)$-refutation in formula space $Sp(\pi_n) = \mathrm{o}\left((n/\log n)^{1/(k+1)}\right)$ must have length $L(\pi_n) = n^{\Omega(\log\log n)}$.*

Having presented trade-off results in the low-space and medium-space range, we conclude by presenting a result at the other end of the space spectrum. Namely, appealing one last time to yet another result in [34], we can show that there are formulas of nearly linear space complexity (recall that any formula is refutable in linear formula space) that exhibit not only superpolynomial but even exponential trade-offs.

We state this final theorem only for standard resolution since it is not clear whether it makes sense for $\mathfrak{R}(k)$. That is, we can certainly derive formal trade-off bounds in terms of the $(k+1)$st square root as in the theorems above, but we do not know whether there actually exist $\mathfrak{R}(k)$-refutation in sufficiently small space so that the trade-offs apply. Hence, such trade-off claims for $\mathfrak{R}(k)$, although impressive looking, might simply be vacuous. We can obtain other exponential trade-offs for $\mathfrak{R}(k)$ (see [15] for the details), but they are not quite as strong as the result below for resolution.

Theorem 11 (Exponential trade-offs for nearly-linear space). *Let κ be any sufficiently large constant. Then there are CNF formulas F_n of size $\Theta(n)$ and width $\mathrm{O}(1)$ and a constant $\kappa' \ll \kappa$ such that:*

- *The formulas F_n have syntactic resolution refutations in total space $\kappa' \cdot n/\log n$.*
- *F_n is also refutable in syntactic resolution in length $\mathrm{O}(n)$ and total space $\mathrm{O}(n)$ simultaneously.*
- *However, any semantic refutation of F_n in clause space at most $\kappa \cdot n/\log n$ has length $\exp\left(n^{\Omega(1)}\right)$.*

To get a feeling for this last trade-off result, note again that the lower bound holds for proof systems with arbitrarily strong derivation rules, as long as they operate with disjunctive clauses. In particular, it holds for proof systems that can in one step derive anything that is semantically implied by the current content of the blackboard. Recall that such a proof system can refute any unsatisfiable CNF formula F with n clauses in length $n+1$ simply by writing down all clauses of F on the blackboard and then concluding, in one single derivation step, the contradictory empty clause implied by F. In Theorem 11 this proof system has space nearly sufficient for such an ultra-short refutation of the whole formula. But even so, when we feed this proof system the formulas F_n and restrict it to having at most $\mathrm{O}(n/\log n)$ clauses on the blackboard at any one given time, it will have to keep going for an exponential number of steps before it is finished.

3 Outline of Proofs

Instead of trying to present the somewhat lengthy formal proofs of our theorems, in this extended abstract we want to focus on providing some intuition for the substitution space theorems that are the keys to our results. Let us first in Section 3.1 discuss the result for standard resolution and describe the proof structure in some detail. The analogous result for k-DNF resolution is proven in a similar way, but with the added technical complications that we need to prove size bounds on sets of k-DNF formulas. These issues, and in particular our result for minimally unsatisfiable sets of k-DNF formulas, are discussed in Section 3.2.

3.1 Proof Ingredients for Substitution Space Theorem for Resolution

Let F be any unsatisfiable CNF formla and f_d any non-authoritarian Boolean function (as described in Section 2.1), and let $F[f_d]$ denote the CNF formula obtained by substituting $f(x_1, \ldots, x_d)$ for every variable x in F and expanding the result to conjunctive normal form.

The first part of Theorem 2, that any resolution refutation of F can be transformed into a refutation of $F[f_d]$ with similar parameters, is not hard to prove. Essentially, whenever the refutation of the original formula F writes a clause C on the blackboard, we write the corresponding set of clauses $C[f_d]$ on the blackboard where we are refuting the substitution formula. We make the additional observation that if we take any resolution refutation π of F and write down the new refutation π_f of $C[f_d]$ resulting from this transformation—assuming for concreteness that the function f_d is exclusive or, say—it is easy to verify that the number of variable occurrences in π, i.e., the *variable space*, translates into a lower bound on the number of clauses in π_f, i.e., the *formula space* (which as we recall is called *clause space* for standard resolution). Equation (1) provides an example of this variable-space-to-clause-space blow-up.

It is more challenging, however, to prove the reverse direction that we can get lower bounds on clause space for $F[f_d]$ from lower bounds on variable space for F. Ideally, we would like to claim that any prover refuting $F[f_d]$ had better write down to the blackboard clause sets on the form $C[f_d]$ corresponding to clauses C in some refutation of the original CNF formula F, and that if he or she does not, then we can analyze the refutation as if that is what is happening anyway, just ignoring the clauses that do not fit into this framework.

To argue this more formally, we need to specify how sets of clauses in a refutation of $F[f_d]$ should be translated to clauses in a purported refutation of F. We do this by devising a way of "projecting" any refutation of $F[f_d]$ down on a refutation of F. These "projections" are defined in terms of a special kind of "precise implication" which we describe next. Recall that for Boolean functions F and G, we say that F *implies* G, denoted $F \vDash G$, if any truth value assignment satisfying F must also satify G.

Definition 12 (Precise implication and projected clauses (informal)). Suppose that $\mathbb{D}$ is a set of clauses over variables in $Vars(F[f_d])$ and that P and N are (disjoint) subset of variables of F. If any truth value assignment satisfying $\mathbb{D}$ must also satisfy $\bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y})$ but this is not the case for strict subsets $P' \subsetneqq P$ or $N' \subsetneqq N$, we say that the clause set $\mathbb{D}$ implies $\bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y})$ *precisely*.

Let us write any clause C as $C = C^+ \vee C^-$, where $C^+ = \bigvee_{x \in Lit(C)} x$ is the disjunction of the positive literals in C and $C^- = \bigvee_{\overline{y} \in Lit(C)} \overline{y}$ is the disjunction of the negative literals. Then we say that $\mathbb{D}$ *projects* C if $\mathbb{D}$ implies $\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})$ precisely, and we write $proj_F(\mathbb{D})$ to denote the set of all clauses that $\mathbb{D}$ projects on yF.

Given this definition, we would like to take any resolution refutation $\pi_f = \{\mathbb{D}_0, \mathbb{D}_1, \ldots, \mathbb{D}_\tau\}$ of $F[f_d]$ and argue that $\pi = \{proj_F(\mathbb{D}_0), proj_F(\mathbb{D}_1), \ldots, proj_F(\mathbb{D}_\tau)\}$ is (essentially) the resolution refutation of F that we are looking for.

It is not hard to see that for a "well-behaved" resolution prover refuting $F[f_d]$ using a resolution refutation π of the original formula F as a template but substituting the clause set $C[f_d]$ for every clause C appearing on the blackboard, applying the projection in Definition 12 at every step in the derivation will give us back the refutation π of F that we started with (the reader can check that this is the case for instance for the clause set in (1)).

What is more remarkable is that this projection of refutations of $F[f_d]$ always works no matter what the prover is doing, in the sense that the result is always a resolution refutation of the original formula F and this projected refutation does not only (essentially) preserve length, which is not too complicated to show, but also space. We refer to [15] for the formal statements and proofs.

3.2 Substitution Space Theorem for $\mathfrak{R}(k)$

The proof of the first part of Theorem 3 is again reasonably straightforward and resembles our proof of the substitution theorem for the standard resolution proof system. For the second part, however, we require a result, described next, that bounds the number of variables appearing in a minimally unsatisfiable k-DNF set of a given size. Since this result addresses a combinatorial problem that appears to be interesting (and challenging) in its own right, we describe it in some detail below.

We start by recalling that a set of 1-DNF formulas, i.e., a CNF formula, is said to be *minimally unsatisfiable* if it is unsatisfiable but every proper subset of its clauses is satisfiable, and try to generalize this definition to the case of $k > 1$. Perhaps the first, naive, idea how to extend this notion is to define $\mathbb{D}$ to be minimally unsatisfiable if it is unsatisfiable but all proper subsets of k-DNF formulas in $\mathbb{D}$ are satisfiable. This will not work, however, and the set of formulas

$$\{x, \big((\overline{x} \wedge y_1) \vee (\overline{x} \wedge y_2) \vee \cdots \vee (\overline{x} \wedge y_n)\big)\} \quad (2)$$

shows why this approach is problematic. The set of formulas (2), which consists of two 2-DNF formulas, is unsatisfiable but every proper subset of it is satisfiable. However, the number of variables appearing in the set can be arbitrarily large so there is no way of bounding $|\mathit{Vars}(\mathbb{D})|$ as a function of $|\mathbb{D}|$.

A more natural requirement is to demand minimality not only at the formula level but also at the term level, saying that not only do all DNF formulas in the set have to be there but also that no term in any formula can be shrunk to a smaller, weaker term without the set becoming satisfiable. Luckily enough, this also turns out to be the concept we need for our applications. The formal definition follows next.

Definition 13 (Minimal implication and minimally unsatisfiable k-DNF sets). Let $\mathbb{D}$ be a set of k-DNF formulas and let G be a formula. We say that $\mathbb{D}$ *minimally implies* G if $\mathbb{D} \vDash G$ and furthermore, replacing any single term T appearing in a single DNF formula $D \in \mathbb{D}$ with a proper subterm of T, and calling the resulting DNF set $\mathbb{D}'$, results in $\mathbb{D}' \nvDash G$. If G is unsatisfiable we say $\mathbb{D}$ is *minimally unsatisfiable*.

To see that this definition generalizes the notion of a minimally unsatisfiable CNF formula, notice that removing a clause C' from a CNF formula F is equivalent to replacing a term of C', which is a single literal, with a proper subterm of it, which is the empty term. This is because the empty term evaluates to 1 on all assignments, which means that the resulting clause also evaluates to 1 on all assignments and hence can be removed from F. With this definition in hand, we are thus interested in understanding the following problem:

> *Given a minimally unsatisfiable set of m k-DNF formulas, what is an upper bound on the number of variables that this set of formulas can contain?*

As was noted above, for $k = 1$ the set $\mathbb{D}$ is equivalent to a CNF formula, because it is a set of disjunctions of literals, and we have the following "folklore" result which seems to have been proved independently on several different occasions (see, for instance, [1, 8, 21, 33]).

Theorem 14. *If $\mathbb{D}$ is a minimally unsatisfiable CNF formula, then $|\mathit{Vars}(\mathbb{D})| < |\mathbb{D}|$.*

Theorem 14 has a relatively elementary proof based on Hall's marriage theorem, but its importance to obtaining lower bounds on resolution length and space is hard to overemphasize. For instance, the seminal lower bound on refutation length of random CNFs given by Chvátal and Szemerédi in [21] makes crucial use of it, as does the proof of the "size-width trade-off" of [16]. Examples of applications of this theorem in resolution space lower bounds include [4, 12, 13, 36, 39].

For sets of k-DNF formulas with $k > 1$, we are not aware of any upper or lower bounds on minimally unsatisfiable sets prior to our work. The main technical result that we need in order to establish the k-DNF resolution space hierarchy is the following extension of Theorem 14 to the case of $k > 1$.

Theorem 15. *Suppose that $\mathbb{D}$ is a minimally unsatisfiable k-DNF set. Then the number of variables in $\mathbb{D}$ is at most $|\mathit{Vars}(\mathbb{D})| \leq (k \cdot |\mathbb{D}|)^{k+1}$.*

Proof sketch. Let us sketch the proof for $k = 2$. Suppose that we have a 2-DNF set $\mathbb{D}$ with m formulas mentioning $\Omega(m^3)$ variables. Then there is at least one 2-DNF formula D^* mentioning $\Omega(m^2)$ variables. By the definition of minimality, the set $\mathbb{D} \setminus \{D\}$ is satisfiable. Let α be some minimal partial assignment fixing $\mathbb{D} \setminus \{D\}$ to true, and note that α needs to set at most $2(m-1)$ variables (at most one 2-term per formula).

Consider the 2-terms in D^*. If there are $2m$ terms over completely disjoint pairs of variables, then there is some 2-term $a \wedge b$ untouched by α. If so, α can be extended to a satisfying assignment for all of $\mathbb{D}$, which is a contradiction. Hence there are at most $\mathrm{O}(m)$ terms over disjoint sets of variables.

But D^* contains $\Omega(m^2)$ variables. By counting (and adjusting the implicit constant factors), there must exist some literal a^* in D^* occuring in a lot of terms $(a^* \wedge b_1) \vee (a^* \wedge b_2) \vee \cdots (a^* \wedge b_{2m})$. Again by minimality, there is a (partial) truth value assignment α' satisfying $\mathbb{D} \setminus \{D\}$ *and setting a^* to true.* (To see this, note that shrinking, for instance, $a^* \wedge b_1$ to a^* should make the whole set satisfiable). But if we pick such an α' of minimal size, there must exist some b_i that is not falsified and we can extend α' to a satisfying assignment for $a^* \wedge b_i$ and hence for the whole set. Contradiction. $\square$

We want to point out that in contrast to Theorem 14, which is exactly tight (consider the set

$$\left\{\bigvee_{i=1}^{n} x_i,\ \neg x_1,\ \neg x_2,\ \ldots,\ \neg x_n\right\} \tag{3}$$

of $n+1$ clauses over n variables), there is no matching lower bound on the number of variables in Theorem 15. The best explicit construction that we were able to obtain, stated next, has number of variables only *linear* in the number of k-DNF formulas (for k constant), improving only by a factor k^2 over the bound for CNF formulas in Theorem 14.

Lemma 16. *There are minimally unsatisfiable k-DNF sets $\mathbb{D}$ with $|\mathit{Vars}(\mathbb{D})| \geq k^2(|\mathbb{D}| - 1)$.*

Proof sketch. Consider any minimally unsatisfiable CNF formula consisting of $n+1$ clauses over n variables (for instance, the one in (3)). Substitute every variable x_i with

$$\begin{aligned} &\left(x_i^1 \wedge x_i^2 \wedge \cdots \wedge x_i^k\right) \\ \vee &\left(x_i^{k+1} \wedge x_i^{k+2} \wedge \cdots \wedge x_i^{2k}\right) \\ \vee &\cdots \\ \vee &\left(x_i^{k^2-k+1} \wedge x_i^{k^2-k+2} \wedge \cdots \wedge x_i^{k^2}\right) \end{aligned} \tag{4}$$

and expand every clause to a k-DNF formula. Note that this is possible since the negation of (4) that we need to substitute for $\neg x_i$ can also be expressed as a k-DNF formula

$$\bigvee_{\substack{(j_1,\ldots,j_k)\in \\ [1,k]\times\ldots\times[(k^2-k+1,k^2]}} \left(\neg x_i^{j_1} \wedge \cdots \wedge \neg x_i^{j_k}\right) \ . \tag{5}$$

It is straightforward to verify that the result is a minimally unsatisfiable k-DNF set in the sense of Definition 13, and this set has $n+1$ formulas over $k^2 n$ variables. $\square$

In our first preliminary report [14] on our results for k-DNF resolution, we stated that we saw no particular reason to believe that the upper bound in Theorem 15 should be tight, hinting that Lemma 16 might well be closer to the truth. Surprisingly to us, this turned out to be wrong. In a joint work [40] with Razborov, the second author recently showed that there are minimally unsatisfiable k-DNF sets with m formulas and $\approx m^k$ variables, which means that Theorem 15 is tight up to an additive one in the exponent.

Concluding this section, we remark that the precise statement required to prove the second part of Theorem 3 is somewhat more involved than Theorem 15. However, the two proofs follow each other very closely. Again, we refer to the full-length version [15] of this paper for the details.

4 Concluding Remarks

We end this paper by discussing some open questions related to our reported work.

Resolution

For the length, width, and clause space measures in resolution, there are known upper and lower worst-case bounds that essentially match modulo constant factors. This is *not* the case for total space, however.

Open Question 1. *Are there polynomial-size CNF formulas of width* $\mathrm{O}(1)$ *which require total resolution refutation space* $TotSp_{\mathfrak{R}}(F \vdash 0) = \Omega\big((\textit{size of } F)^2\big)$*?*

The answer has been conjectured by [4] to be "yes", but as far as we are aware, there are no stronger lower bounds on total space known than those that follow trivially from corresponding linear lower bounds on clause space. Thus, a first step would be to show superlinear lower bounds on total space.

One way of interpreting the results of the current paper is that time-space trade-offs in pebble games carry over more or less directly to the resolution proof system (modulo some technical restrictions that we ignore here). The resolution trade-off results obtainable by this method are inherently limited, however, in the sense that pebblings in small space can be seen never to take too much time by a simple counting argument. For resolution there are no such limitations, at least not a priori, since the corresponding counting argument does not apply. Thus, one can ask whether it is possible to demonstrate even more dramatic time-space trade-offs for resolution than those that can be obtained via pebbling.

To be more specific, we are particularly interested in what trade-offs are possible at the extremal points of the space interval, where we can only get polynomial trade-offs for constant space and no trade-offs at all for linear space.

Open Question 2. *Are there superpolynomial trade-offs for formulas refutable in constant clause space?*

Open Question 3. *Are there formulas with trade-offs in the range space* $>$ *formula size? Or can every resolution refutation be carried out in at most linear space?*

We find Open Question 3 especially intriguing. Note that all bounds on clause space proven so far, inlcuding the trade-offs in the current paper, are in the regime where the space is less than formula size (which is quite natural, since by [25] we know the size of the formula is an upper bound on the minimal clause space needed). It is unclear to what extent such lower bounds on space are relevant to state-of-the-art SAT solvers, however, since such algorithms will presumably use at least a linear amount of memory to store the formula to begin with. For this reason, it seems to be a highly interesting problem to determine what can be said if we allow extra clause space above linear. Are there formulas exhibiting trade-offs in this superlinear regime, or is it always possible to carry out a minimal-length refutation in, say, at most a constant factor times the linear upper bound on the space required for any formula? As was noted above, pebbling formulas cannot help answer these two questions, since they are always refutable in linear time and linear space simultaneously by construction, and since constant pebbling space implies polynomial pebbling time.

A final problem related specifically to standard resolution is that it would be interesting to investigate the implications of our results for applied satisfiability algorithms.

Open Question 4. *Do the trade-off phenomena we have established in this paper show up "in real life" for state-of-the-art DPLL based SAT-solvers, when run on the appropriate pebbling contradictions (or variations of such pebbling contradictions)?*

Stronger Space Separations for $\mathfrak{R}(k)$

We have proven a strict separation between k-DNF resolution and $(k+1)$-DNF resolution by exhibiting for every fixed k a family of CNF formulas of size n that require space $\Omega\big(\sqrt[k+1]{n/\log n}\big)$ for any k-DNF resolution refutation but can be refuted in constant space in $(k+1)$-DNF resolution. This shows that the family of $\mathfrak{R}(k)$ proof systems form a strict hierarchy with respect to space.

As has been said above, however, we have no reason to believe that the lower bound for $\mathfrak{R}(k)$ is tight. In fact, it seems reasonable that a tighter analysis should be able to improve the bound to at least $\Omega\big(\sqrt[k]{n/\log n}\big)$ and possibly even further. The only known *upper* bound on the space needed in $\mathfrak{R}(k)$ for these formulas is the $\mathrm{O}(n/\log n)$ bound that is easily obtained for standard resolution. Closing, or at least narrowing, the gap between $\Omega\big(\sqrt[k+1]{n/\log n}\big)$ and $\mathrm{O}(n/\log n)$ is hence an open question.

Understanding Minimally Unsatisfiable Sets of k-DNF Formulas

It seems that the problem of getting better lower bounds on space for k-DNF resolution is related to the problem of better understanding the structure of minimally unsatisfiable sets of k-DNF formulas. Although the correspondence is more intuitive than formal, it would seem that progress on this latter problem would probably translate into sharper lower bounds for $\mathfrak{R}(k)$ as well. The reason for this hope is that the asymptotically optimal results for standard resolution in this paper can in some sense be seen to follow from (the proof technique used to obtain) the tight bound for CNF formulas in Theorem 14.

What we are able to prove in this paper is that any minimally unsatisfiable k-DNF set $\mathbb{D}$ (for k a fixed constant) must have at least $\Omega\big(\sqrt[k+1]{|\mathbb{D}|}\big)$ variables (Theorem 15) but the only explicit constructions of such sets that we where able to obtain had $\mathrm{O}(|\mathbb{D}|)$ variables (Lemma 16). As has already been mention, the recent work [40] unexpectedly improved the lower bound to roughly $\mathrm{O}\big(\sqrt[k]{|\mathbb{D}|}\big)$. This appears to be a natural and interesting combinatorial problem in its own right, and it would be very nice to close the gap between the upper and lower bound.

We have the following conjecture, where for simplicity we fix k to remove it from the asymptotic notation.

Conjecture 17. *Suppose that $\mathbb{D}$ is a minimally unsatisfiable k-DNF set for some arbitrary but fixed positive integer k. Then the number of variables in $\mathbb{D}$ is at most $\mathrm{O}(|\mathbb{D}|)^k$.*

Proving this conjecture would establish asymptotically tight bounds for minimally unsatisfiable k-DNF sets (ignoring factors involving the constant k).

Generalizations to Other Proof Systems

We have presented a "substitution space theorem" for resolution as a way of lifting lower bounds on the number of variables to lower bounds on (clause) space, and have then extended this result by lifting lower bounds on the number of variables *in resolution* to lower bounds on formula space in the *much stronger k-DNF resolution proof systems*. It is a natural question to ask whether our techniques can be extended to other proof systems as well.

We remark that our translations of refutations of substitution formulas in some other proof system $\mathcal{P}$ via projection to resolution refutations of the original formula seem extremely generic and robust in that they do not at all depend on which derivation rules are used by $\mathcal{P}$ nor on the class of formulas with which $\mathcal{P}$ operates. The only place where the particulars of the proof system come into play is when we actually need to analyze the content of the proof blackboard. As described in the introduction, this happens at some critical point in time when we know that the blackboard of our translated (projected) resolution proof mentions a lot of variables, and want to argue that this implies that the blackboard of the $\mathcal{P}$-proof must contain a lot of formulas (or possibly some other resource that we want to lower-bound in $\mathcal{P}$). Any corresponding result for some other proof system $\mathcal{P}$ would translate into lower bounds for $\mathcal{P}$ in terms of lower bounds on variable space in resolution.

Acknowledegments

The research of the first author leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 240258 and was supported by the Israeli Science Foundation and by the US-Israel Binational Science Foundation under grant number 2006104.

The research of the second author was performed while at the Massachusetts Institute of Technology supported by grants from the Royal Swedish Academy of Sciences, the Ericsson Research Foundation, the Sweden-America Foundation, the Foundation Olle Engkvist Byggmästare, the Sven and Dagmar Salén Foundation, and the Foundation Blanceflor Boncompagni-Ludovisi, née Bildt.

References

[1] R. Aharoni and N. Linial. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *Journal of Combinatorial Theory*, 43: 196–204, 1986.

[2] M. Ajtai. The complexity of the pigeonhole principle. In *Proceedings of the 29th Annual IEEE*

Symposium on Foundations of Computer Science (FOCS '88), pages 346–355, Oct. 1988.

[3] M. Alekhnovich. Lower bounds for k-DNF resolution on random 3-CNFs. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pages 251–256, May 2005.

[4] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002. Preliminary version appeared in *STOC '00*.

[5] A. Atserias and M. L. Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189(2): 182–201, Mar. 2004. Preliminary version appeared in *CSL '02*.

[6] A. Atserias and V. 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*.

[7] A. Atserias, M. L. Bonet, and J. L. Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Information and Computation*, 176(2):136–152, Aug. 2002. Preliminary version appeared in *ICALP '01*.

[8] S. Baumer, J. L. Esteban, and J. Torán. Minimally unsatisfiable CNF formulas. *Bulletin of the European Association for Theoretical Computer Science*, 74:190–192, June 2001.

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

[10] P. Beame, T. Huynh, and T. Pitassi. Hardness amplification in proof complexity. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC '10)*, pages 87–96, June 2010.

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

[12] E. Ben-Sasson and N. Galesi. Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, Aug. 2003. Preliminary version appeared in *CCC '01*.

[13] E. Ben-Sasson and J. Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 709–718, Oct. 2008.

[14] E. Ben-Sasson and J. Nordström. A space hierarchy for k-DNF resolution. Technical Report TR09-047, Electronic Colloquium on Computational Complexity (ECCC), Apr. 2009.

[15] E. Ben-Sasson and J. Nordström. Understanding space in proof complexity: Separations and tradeoffs via substitutions. Technical Report TR10-125, Electronic Colloquium on Computational Complexity (ECCC), Aug. 2010.

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

[17] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson. Near optimal separation of treelike and general resolution. *Combinatorica*, 24(4):585–603, Sept. 2004.

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

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

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

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

[22] S. A. Cook and R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, Mar. 1979.

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

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

[25] J. L. Esteban and J. Torán. Space bounds for resolution. *Information and Computation*, 171(1): 84–97, 2001. Based on the conference papers in *STACS '99* [26] and *CSL '99* [45].

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

[27] J. L. Esteban, N. Galesi, and J. Messner. On the complexity of resolution with bounded conjunctions. *Theoretical Computer Science*, 321(2-3):347–370, Aug. 2004. Preliminary version appeared in *ICALP '02*.

[28] J. R. Gilbert and R. E. Tarjan. Variations of a

pebble game on graphs. Technical Report STAN-CS-78-661, Stanford University, 1978. Available at the webpage http://infolab.stanford.edu/TR/CS-TR-78-661.html.

[29] P. Hertel and T. Pitassi. Exponential time/space speedups for resolution and the PSPACE-completeness of black-white pebbling. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*, pages 137–149, Oct. 2007.

[30] P. Hertel and T. Pitassi. The PSPACE-completeness of black-white pebbling. *SIAM Journal on Computing*, 39(6):2622–2682, Apr. 2010. Preliminary version appeared in *FOCS '07*.

[31] J. Johannsen and N. S. Narayanaswamy. An optimal lower bound for resolution with 2-Conjunctions. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science (MFCS '02)*, volume 2420 of *Lecture Notes in Computer Science*, pages 387–398. Springer, Aug. 2002.

[32] J. Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3): 123–140, 2001.

[33] O. Kullmann. An application of matroid theory to the SAT problem. In *Proceedings of the 15th Annual IEEE Conference on Computational Complexity (CCC '00)*, pages 116–124, July 2000.

[34] T. Lengauer and R. E. Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *Journal of the ACM*, 29(4):1087–1130, Oct. 1982. Preliminary version appeared in *STOC '79*.

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

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

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

[38] J. Nordström. On the relative strength of pebbling and resolution (Extended abstract). In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC '10)*, pages 151–162, June 2010.

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

[40] J. Nordström and A. Razborov. On minimal unsatisfiability and time-space trade-offs for k-DNF resolution. Technical Report TR09-100, Electronic Colloquium on Computational Complexity (ECCC), Oct. 2009.

[41] A. A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. Manuscript. Available at the webpage http://people.cs.uchicago.edu/~razborov/research.html, 2002-2003.

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

[43] N. Segerlind. Exponential separation between $\mathrm{Res}(k)$ and $\mathrm{Res}(k+1)$ for $k \leq \epsilon \log n$. *Information Processing Letters*, 93(4):185–190, Feb. 2005.

[44] N. Segerlind, S. R. Buss, and R. Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM Journal on Computing*, 33(5):1171–1200, 2004. Preliminary version appeared in *FOCS '02*.

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