A Space Hierarchy for k-DNF Resolution*

Eli Ben-Sasson†
Computer Science Department
Technion — Israel Institute of Technology
Haifa, 32000, Israel
eli@cs.technion.ac.il

Jakob Nordström‡
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
jakobn@mit.edu

April 20, 2009

Abstract

The k-DNF resolution proof systems are a family of systems indexed by the integer k, where the k^{th} member is restricted to operating with formulas in disjunctive normal form with all terms of bounded arity k (k-DNF formulas). This family was introduced in [Krajíček 2001] as an extension of the well-studied resolution proof system. A number of lower bounds have been proven on k-DNF resolution proof length and space, and it has also been shown that $(k+1)$-DNF resolution is exponentially more powerful than k-DNF resolution for all k with respect to length. For proof space, however, no corresponding hierarchy has been known except for the (very weak) subsystems restricted to tree-like proofs. In this work, we establish a strict space hierarchy for the general, unrestricted k-DNF resolution proof systems.

1 Introduction

Proof space A central theme in the field of propositional proof complexity is the study of limitations of natural proof systems. This is typically done by considering a *complexity measure* of propositional proofs and studying under which circumstances this measure is large. The most heavily investigated complexity measure is that of *proof size/length* and the interest in this measure is motivated by its connections to the NP vs. co-NP problem (see [CR79] for details), to methods for proving independence results in first order theories of bounded arithmetic (for an example, see [Ajt88]), 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*).

This paper focuses on a more recently suggested complexity measure known as *space*. The space measure was first defined and studied by Esteban and Torán [ET01] in the context of the famous *resolution*

*This work was initiated during the seminar on Computational Complexity of Discrete Problems held at Schloss Dagstuhl in September 2008 (seminar 08381).

†Research supported by the Israeli Science Foundation and by the US-Israel Binational Science Foundation.

‡Research supported by the Ericsson Research Foundation, the Foundation Olle Engkvist Byggmästare, and the Foundation Blanceflor Boncompagni-Ludovisi, née Bildt.

proof system introduced by Blake [Bla37], and was generalized to other proof systems by Alekhnovich et al. in [ABSRW02]. Roughly speaking, the space of proving a formula corresponds to the minimal size of a blackboard needed to verify all steps in the proof. 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. Esteban and Torán showed in [ET01] that space lower bounds imply length lower bounds for the proof system 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 connection between space and length is that these two measures sometimes display a *trade-off*. By this we mean that there are formulas having proofs in both short length and small space, but for which there cannot exist proofs in short length and small space *simultaneously*. Such a space-length trade-off was shown initially for tree-like resolution by the first author in [BS02] and more recently for (non-tree-like) resolution in [HP07, Nor07, BSN09].

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 [DP60, DLL62] and known as *Davis-Putnam-Logemann-Loveland (DPLL)* SAT solvers have the following property. When given as input an unsatisfiable formula F in conjunctive normal form—called henceforth a *CNF formula*—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. Of late, a family of SAT solvers known as *DPLL with clause learning* (denoted 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 (see [SAT]) 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, unrestricted resolution translate into memory lower bounds for these algorithms.

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 far greater than the theoretical upper bounds on refutation space. On the other hand, the theoretical lower bounds on refutation space are *worst-case* bounds for *non-deterministic algorithms*, i.e., they apply even to the most memory-efficient proof theoretically possible, which is not remotely close to the kind of proofs produced by a typical SAT solver. Understanding what kind of limitations one can get on the memory consumption of SAT solvers from refutation space lower bounds remains as an interesting challenge.

k-DNF resolution The family of sequential proof systems known as *k-DNF resolution* was introduced by Krajíček in [Kra01] as a intermediate step between resolution and depth-2 Frege. Roughly speaking, for integers $k > 0$ the k^{th} 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 system $\mathfrak{R}(1)$ is 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 [Kra01], 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 [JN02] for a discussion of this point). By now a number of works have shown superpolynomial lower bounds on the length of

[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* (See Section 2 for formal definitions).

$\mathfrak{R}(k)$-refutations, most notably for (various formulations of) the pigeonhole principle and for random CNF formulas [AB04, ABE02, Ale05, JN02, Raz03, SBI04, Seg05]. Of particular relevance to our current work are the results of Segerlind et al. [SBI04] and of Segerlind [Seg05] showing that the family of $\mathfrak{R}(k)$ systems form a *strict hierarchy* with respect to proof length. More precisely, they prove that for every integer $k > 0$ there exists a family of formulas $\{F_n\}$ 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 $2^{\Omega(n)}$.

Just as in the case for standard resolution, the understanding of space complexity in $\mathfrak{R}(k)$ has remained more limited. We are aware of only one prior work by Esteban et al. [EGM04] shedding light on this question. Their paper establish essentially optimal 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 [Kra01, EGM04] 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 [EGM04] leave completely open the question of whether there is a strict space hierarchy for (non-tree-like) $\mathfrak{R}(k)$ or not.

Main result—a space hierarchy for k-DNF resolution Our main result is that Krajíček's family of $\mathfrak{R}(k)$ systems do indeed form a strict hierarchy with respect to space. To explain this result we need to describe more formally what we mean by "space". We view an unsatisfiable CNF formula F as a set of clauses and, following [ABSRW02], define a $\mathfrak{R}(k)$-refutation of F a to be a sequence of *sets* of k-DNF formulas $\pi = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ such that $\mathbb{D}_0$ is the empty set and $\mathbb{D}_\tau$ contains the contradictory empty formula. Informally, $\mathbb{D}_t$ is a snapshot of the blackboard after the t^{th} step of the proof has been performed. The allowed steps, i.e., the transitions from $\mathbb{D}_{t-1}$ to $\mathbb{D}_t$ deemed as legal, correspond to *(i)* writing a clause of F on the blackboard, *(ii)* erasing a line from the board, and *(iii)* inferring a new line from those lines present on the board according to the inference rules of $\mathfrak{R}(k)$.[2]

The length of a refutation is the number of derivation steps in it. 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 lines on the board, i.e., to measure the size of $\mathbb{D}_t$, denoted $|\mathbb{D}_t|$. We call this the *formula space*, or simply, *space* of $\mathbb{D}_t$. (For standard resolution, this is the well-studied measure of *clause space*.) A finer granulation is to measure the *term space*—the number of terms appearing in the formulas of $\mathbb{D}$. Viewing a DNF formula D as a set of terms this measure is $\sum_{D \in \mathbb{D}_t} |D|$. An even finer measure is the *variable space*—the number of appearances of literals in $\mathbb{D}_t$, counted with repetition. Viewing a term T as a set of literals this is $\sum_{D \in \mathbb{D}_t} \sum_{T \in D} |T|$.

Our hierarchy theorem says that for every fixed k there exists a family of *efficiently constructible*[3] unsatisfiable CNF formulas $\{F_n\}_{n=1}^{\infty}$ such that any $\mathfrak{R}(k)$-refutation of F_n must have (formula) space at least $\Omega\bigl(\sqrt[k+1]{n/\log n}\bigr)$ but on the other hand F_n can be refuted in $\mathfrak{R}(k+1)$ in constant variable space. (Moreover, the constant space $\mathfrak{R}(k+1)$-refutation of F_n is also of linear length in n.) We point out that these bounds in fact hold for all space measures discussed above, since the upper bound on space is in terms of the largest of the space measures defined above—variable space, whereas the lower bound is stated in terms of the smallest of these measures—formula space.

Minimally unsatisfiable k-DNF sets We end our overview by focusing on the main technical novelty of this paper which discusses a question that may be of independent interest. The upper bound on the variable

[2] These rules are given in Definition 2.1, but for our results the exact definitions in fact do not matter—our lower bounds 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.

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

space of refuting F_n in $\mathfrak{R}(k+1)$ carries over quite straightforwardly from our recent work [BSN09]. It is the lower bound on the formula space in $\mathfrak{R}(k)$ that requires a new idea.

Namely, in our proof of the lower bound we reach a crucial point where we have on the k-DNF resolution "proof blackboard" a set $\mathbb{D}$ of k-DNF formulas that involves very many variables, but implies[4] a number of small (and strong) formulas G over very few variables. We wish to conclude that the only way this can happen is that the blackboard set $\mathbb{D}$ contains many distinct formulas. For the sake of simplicity let us assume G is itself unsatisfiable, i.e., G is a formula computing the constant 0 function. Saying "$\mathbb{D}$ implies 0" is equivalent to saying that $\mathbb{D}$ is unsatisfiable, i.e., there is no assignment that satisfies all k-DNF formulas in $\mathbb{D}$. Assuming $\mathbb{D}$ is unsatisfiable and involves at least n variables, can we bound from below the size of $\mathbb{D}$? As stated, the answer to this question is a flat "no". To see this consider the following unsatisfiable set consisting of three k-DNF formulas, each formula involving a single term:

$$\left\{x \wedge \bigwedge_{i=1}^{n-1} y_i,\ \neg x \wedge \bigwedge_{i=1}^{n-1} y_i,\ \bigwedge_{i=1}^{n-1} y_i\right\} \tag{1}$$

Even if we "weaken" any term (i.e., make it easier to satisfy) by removing from it a variable labeled y_i, the remaining set is unsatisfiable. The reason for this is that the set above "hides" within it a weaker set of formulas that is already unsatisfiable, namely, the set $\{x, \neg x\}$. This suggests rephrasing our question as follows. We say that a set $\mathbb{D}$ of k-DNF formulas is *minimally unsatisfiable* if weakening any single term appearing in it will make the "weaker" set of formulas satisfiable.

Open Problem 1. *What is the minimal size of a set of k-DNF formulas that is minimally unsatisfiable and mentions n variables?*

For $k = 1$ this question has been completely resolved. In this case, $\mathbb{D}$ 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.

Theorem 1.1 ([AL86, BET01, CS88, Kul00]). *If $\mathbb{D}$ is a set of 1-DNF formulas, i.e., a CNF formula, that is minimally unsatisfiable and mentions n variables, then $|\mathbb{D}| > n$.*

The following minimally unsatisfiable set of $n+1$ clauses over n variables shows that the bound stated above is tight.

$$\left\{\bigvee_{i=1}^{n} x_i,\ \neg x_1,\ \neg x_2,\ \ldots,\ \neg x_n\right\} \tag{2}$$

Theorem 1.1 has a relatively elementary proof based on Hall's marriage theorem, but its importance to obtaining lower bounds on resolution length and space cannot be overemphasized. For instance, the seminal lower bound on refutation length of random CNFs given by Chvátal and Szemerédi in [CS88] makes crucial use of it, as does the proof of the "size-width trade-off" of [BSW01]. Examples of applications of this theorem in resolution space lower bounds include [ABSRW02, BSG03, BSN08, BSN09, NH08, Nor06].

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 an extension of the lower bound in Theorem 1.1 to the case of $k > 1$. Our result, stated in Theorem 3.5, says that a minimally unsatisfiable set of k-DNF formulas involving n variables must have size at least $\sqrt[k+1]{n/k}$. Notice that the lower bound on the size of a minimally unsatisfiable set of k-DNF formulas is of the same asymptotic order as the $\mathfrak{R}(k)$-space lower bound stated above. We point out that the result needed for our space lower bound, stated in Lemma 4.3, has to work for a more general definition of "minimal implication" (see Section 4 for more details). However, it seems reasonable to believe that improving the bound stated in Theorem 3.5 for the more restricted Problem 1 would lead also to a stronger space separation of $\mathfrak{R}(k)$ and $\mathfrak{R}(k+1)$. We end by stating that we do not see any reason to believe our lower bound for $k > 1$ is asymptotically tight. In fact, we are not aware of any sets of minimally unsatisfiable k-DNF formulas that are of size $\mathrm{o}(n)$.

[4] A set of formulas $\mathbb{D}$ *implies* a formula G if and only if every assignment that satisfies all formulas in $\mathbb{D}$ must also satisfy G.

Organization of the rest of the paper After presenting the formal definitions in Section 2, we give precise statements of our main results in Section 3. Section 4 starts the proof of the $\mathfrak{R}(k)$-space lower bound on F_n. Section 5 is the technical heart of the paper and studies the size of minimally unsatisfiable formula sets. In Section 6, we complete the proof of the space lower bound for $\mathfrak{R}(k)$. Section 7 provides the final missing component, namely the (relatively straightforward) upper bound on the $\mathfrak{R}(k+1)$-refutation space of F_n. We conclude in Section 8 with a brief discussion of directions for future research.

2 Preliminaries

In this section we give the formal definitions used in this paper and state a few basic facts that we will need.

2.1 Formulas

For the most part we stick with the standard notations for formulas in conjunctive normal form (CNF) and disjunctive normal form (DNF). However, often we will use the very convenient, although somewhat less standard, set notation to treat objects such as clauses, terms, restrictions and CNF and DNF formulas. We explain this terminology next. Definitions of standard notation regarding formulas can be found in, for instance, [Nor08, Section 4.4]. Throughout this paper, we let $[k]$ denote the set $\{1, \ldots, k\}$.

DNF and CNF formulas as sets For x a Boolean variable, a *literal over* x is either a Boolean variable x, called a *positive literal over* x or its negation, denoted $\neg x$ or $\overline{x}$ and called a *negative literal over* x. We define $\neg\neg x$ to be x. When x is understood from context or unimportant we simply speak of a (positive, negative) *literal*. A *CNF formula* is a set of clauses, i.e., disjunctions of literals, and a *DNF formula* is a set of terms, i.e., conjunctions of literals. The *variable set* of a term T, denoted $\mathit{Vars}(T)$, is the set of Boolean variables over which there are literals in T. The variable set of a clause is similarly defined and this definition is extended to CNF and DNF formulas by taking unions, i.e., for $D = \{D_1, \ldots, D_s\}$ a DNF formula we define its variable set as $\mathit{Vars}(D) = \bigcup_{i=1}^{s} \mathit{Vars}(D_i)$. If X is a set of Boolean variables and $\mathit{Vars}(T) \subseteq X$ we say T is a term *over* X and similarly define clauses, CNF formulas, and DNF formulas over X.

We think of a *clause* as a set of literals and so is a *term*. We will sometimes borrow set-theoretic notation and terminology to discuss logical formulas. For instance, we say that the term T' is a *subterm* of T, and write $T' \subseteq T$ to denote that the set of literals of T' is contained in the set of literals of T. We similarly speak of, and denote, subclauses and subformulas. We say the clause C (or term T) is a *k-clause* (*k-term*, respectively) if $|C| \leq k$ ($|T| \leq k$, respectively). A *k-DNF formula* D is a set of k-terms and a *k-CNF formula* is a set of k-clauses. The *size* of a DNF formula D, denoted $|D|$, is the number of terms in it and the size of CNF formula is analogously denoted and defined.

Assignments and restrictions as sets As is the case with CNF and DNF formulas, we prefer to use in our proof a set-theoretic representation of restrictions and assignments, as defined next.

A *restriction* ρ over a set of Boolean variables X is a subset of literals over X with the property that for each variable $x \in X$ there is at most one literal over x in ρ. The *set of variables assigned* by ρ is $\mathit{Vars}(\rho)$ and the *size* of ρ is $|\rho| = |\mathit{Vars}(\rho)|$. We say the restriction ρ' *extends* ρ if $\rho' \supseteq \rho$, and in this case we also say that ρ *agrees* with ρ'. An *assignment* α to X is a restriction satisfying $|\alpha| = |X|$.

For a a literal over X and ρ a restriction over X, let the restriction of a under ρ be

$$a\restriction_\rho = \begin{cases} 1 & a \in \rho \\ 0 & \neg a \in \rho \\ a & \text{otherwise} \end{cases} \tag{3}$$

If $a\lceil_\rho = 1$ we say ρ *satisfies* a, if $a\lceil_\rho = 0$ we say ρ *falsifies* a and otherwise we say ρ leaves a *unfixed*. We extend the definition of a restriction to a term $T = a_1 \wedge \cdots \wedge a_s$ and clause $C = a'_1 \vee \cdots \vee a'_s$ as follows. Let $\neg\rho = \{\neg a | a \in \rho\}$ denote the restriction obtained by replacing every literal in ρ by its negation.

$$T\lceil_\rho = \begin{cases} 1 & T \subseteq \rho \\ 0 & T \cap \neg\rho \neq \emptyset \\ T \setminus \rho & \text{otherwise} \end{cases}, \quad C\lceil_\rho = \begin{cases} 0 & C \subseteq \neg\rho \\ 1 & C \cap \rho \neq \emptyset \\ C \setminus \neg\rho & \text{otherwise} \end{cases} \tag{4}$$

In words, we say T is *satisfied* by ρ if ρ satisfies all literals in T, we say T is *falsified* by ρ if some literal of ρ is falsified and otherwise T is unfixed by ρ. Dually, C is satisfied if some literal of it is satisfied by ρ, it is falsified if all its literals are falsified by ρ and otherwise it remains unfixed. Notice that the *empty term*, i.e., the term of size 0, is satisfied by every restriction and the empty clause is falsified by all of them. We extend the definition of a restriction to a DNF formula $D = D_1 \vee \cdots \vee D_m = \{D_1, \ldots, D_m\}$ by

$$D\lceil_\rho = \begin{cases} 1 & \exists i \in [m], D_i\lceil_\rho = 1 \\ 0 & D_i\lceil_\rho = 0, i \in [m] \\ \{D_i\lceil_\rho \;:\; D_i\lceil_\rho \neq 0\} & \text{otherwise,} \end{cases} \tag{5}$$

and to a CNF formula $F = C_1 \wedge \cdots \wedge C_m = \{C_1, \ldots, C_m\}$ by

$$F\lceil_\rho = \begin{cases} 0 & \exists i \in [m], C_i\lceil_\rho = 0 \\ 1 & C_i\lceil_\rho = 1, i \in [m] \\ \{C_i\lceil_\rho \;:\; C_i\lceil_\rho \neq 1\} & \text{otherwise.} \end{cases} \tag{6}$$

The notions of a restriction satisfying, falsifying and leaving unfixed a DNF or CNF formula are analogous to those defined for terms and clauses. If ρ is a restriction satisfying a formula F, yet every proper subrestriction $\rho' \subsetneq \rho$ does not satisfy F, then we say ρ is a *minimal* satisfying restriction. A minimal falsifying restriction is analogously defined. When α is an assignment and F is a formula we use the standard notation of $F(\alpha)$ to denote $F\lceil_\alpha$.

A term (clause, respectively) is said to be *trivial* if it contains both a positive and a negative literal over the same variable. We may assume without loss of generality that all terms (clauses, respectively) appearing in our paper are nontrivial, because the value of a DNF (CNF, respectively) remains unchanged after addition or removal of trivial terms (clauses, respectively). We say that a DNF formula D over X *represents* a Boolean function $f : X \to \{0,1\}$ if and only if for all assignments $\alpha \in \{0,1\}^X$, we have $f(\alpha) = D(\alpha)$. The notion of a CNF formula representing f is analogously defined. It is well-known that every Boolean function can be represented by a CNF and by a DNF.

Implication If $\mathbb{C}$ is a set of formulas we say that a restriction (or assignment) *satisfies* $\mathbb{C}$ if and only if it satisfies every formula in $\mathbb{C}$. For $\mathbb{D}, \mathbb{C}$ two sets of formulas over a set of variables X, we say that $\mathbb{D}$ *implies* $\mathbb{C}$, denoted $\mathbb{D} \vDash \mathbb{C}$, if and only if every assignment α to X that satisfies $\mathbb{D}$ also satisfies $\mathbb{C}$. In particular, $\mathbb{D} \vDash 0$ if and only if $\mathbb{D}$ is *unsatisfiable*, i.e., no assignment satisfies $\mathbb{D}$.

2.2 k-DNF Resolution

We now give a more precise description of the k-DNF resolution proof systems and the proof complexity measures for these systems that we are interested in studying.

Definition 2.1 (k-DNF-resolution inference rules). The *k-DNF-resolution* proof systems are a family of sequential proof systems parameterized by $k \in \mathbb{N}^+$. Lines in a k-DNF-resolution refutation are k-DNF formulas and the following inference rules are allowed (where A, B, C denote k-DNF formulas, T, T' denote k-terms, and $a_1, \ldots, a_k$ denote literals):

k-cut $\frac{(a_1 \wedge \ldots \wedge a_{k'}) \vee B, \quad \neg a_1 \vee \ldots \vee \neg a_{k'} \vee C}{B \vee C}$, where $k' \leq k$.

$\wedge$-introduction $\frac{A \vee T, \quad A \vee T'}{A \vee (T \cup T')}$, as long as $|T \cup T'| \leq k$.

$\wedge$-elimination $\frac{A \vee T}{A \vee T'}$ for any $T' \subseteq T$.

Weakening $\frac{A}{A \vee B}$ for any k-DNF formula B.

The formulas above the line are called the *inference assumptions* and the formula below is called the *consequence*. For brevity we denote by $\mathfrak{R}(k)$ the proof system of k-DNF resolution.

The following definition is the straightforward generalization to $\mathfrak{R}(k)$ of the space-oriented definition of a refutation from [ABSRW02].

Definition 2.2 (Derivation). A *k-DNF configuration* $\mathbb{D}$, or, simply, a *configuration*, is a set of k-DNF formulas. A sequence of configurations $\{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ is said to be a $\mathfrak{R}(k)$*-derivation* from a CNF formula F if $\mathbb{D} = \emptyset$ and for all $t \in [\tau]$, the set $\mathbb{D}_t$ is obtained from $\mathbb{D}_{t-1}$ by one of the following *derivation steps*:

Axiom Download $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{C\}$ for some $C \in F$.

Inference $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{D\}$ for some D inferred by one of the inference rules listed in Definition 2.1 from a set of assumptions that belongs to $\mathbb{D}_{t-1}$.

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

A $\mathfrak{R}(k)$-derivation $\pi : F \vdash \mathbb{D}'$ of a k-DNF set $\mathbb{D}'$ from a formula F is a derivation $\pi = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ such that $\mathbb{D}_\tau = \mathbb{D}'$. A $\mathfrak{R}(k)$*-refutation* of F is a $\mathfrak{R}(k)$-derivation of the empty DNF (denoted by 0), i.e., the DNF formula with no terms, or, phrased differently, the unsatisfiable empty disjunction.

When the derived k-DNF set $\mathbb{D}'$ contains a single formula D, we will often abuse notation slightly by writing simply $\pi : F \vdash D$ instead of $\pi : F \vdash \{D\}$.

Definition 2.3 (Refutation length and space). The *formula space*, or simply *space*, of a configuration $\mathbb{D}$ is its size $|\mathbb{D}|$. The *variable support size*, or just *support size*, of $\mathbb{D}$, denoted $\mathit{SuppSize}(\mathbb{D})$, is the number of variables appearing in $\mathbb{D}$, i.e., $\mathit{SuppSize}(\mathbb{D}) = |\mathit{Vars}(\mathbb{D})|$ and the *variable space* of $\mathbb{D}$, denoted $\mathit{VarSp}(\mathbb{D})$ is the number of variables appearing in $\mathbb{D}$ counted with repetitions. (Notice that $\mathit{VarSp}(\mathbb{D}) \geq \mathit{SuppSize}(\mathbb{D})$.)

The *length* of a $\mathfrak{R}(k)$-derivation π is the number of axiom downloads and inference steps in it. The space (support size, variable space, respectively) of a derivation π is defined as the maximal space (support size, variable space, respectively) of a configuration in π. If π is a derivation of $\mathbb{D}$ from a formula F of length L and space s then we say $\mathbb{D}$ can be derived from F in length L and space s *simultaneously*.

We define the $\mathfrak{R}(k)$*-refutation length* of a formula F, denoted $L_{\mathfrak{R}(k)}(F \vdash 0)$, to be the minimum length of any $\mathfrak{R}(k)$-refutation of it. The $\mathfrak{R}(k)$*-refutation space* of F, denoted $\mathit{Sp}_{\mathfrak{R}(k)}(F \vdash 0)$, and the $\mathfrak{R}(k)$*-refutation support size* of F, denoted $\mathit{SuppSize}_{\mathfrak{R}(k)}(F \vdash 0)$, are analogously defined by taking minima over all $\mathfrak{R}(k)$-refutations of F.

When the proof system $\mathfrak{R}(k)$ in question is clear from context, we will drop the subindex in the proof complexity measures.

Notice that the system $\mathfrak{R}(1)$ is the usual *resolution* proof system. We remark that in resolution, the $\wedge$-introduction and $\wedge$-elimination rules do not apply, and the cut rule reduces to the familiar *resolution rule* saying that the clauses $C_1 \vee x$ and $C_2 \vee \neg x$ can be combined to derive $C_1 \vee C_2$. Also, although the weakening rule is sometimes convenient for technical reasons, it is easy to show that any weakening steps can always be eliminated from a standard resolution refutation of an unsatifiable CNF formula without changing anything essential. Thus, while the results in this paper will be stated for resolution with the weakening rule, they also hold for resolution refutations using only axiom downloads, resolution rule applications and erasures. Let us highlight this fact in a (somewhat) formal proposition for the record.

Proposition 2.4 (Weakenings can be eliminated from resolution refutations). *Any resolution refutation $\pi : F \vdash 0$ using the weakening rule can be transformed into a refutation $\pi' : F \vdash 0$ without weakening such that π' performs at most the same number of axiom downloads, inferences and erasures as does π, and such that for any length or space complexity measure M studied in this paper*[5] *it holds that $M(\pi') \leq M(\pi)$.*

The proof of Proposition 2.4 is an easy forward induction over the resolution refutation (simply ignoring all weakening moves and keeping the subclauses instead, which can never increase neither length nor space). We omit the details.

We will also make use of the *implicational completeness* of resolution. Formally, this means that if $\mathbb{C}$ is a set of clauses and C is a clause, then $\mathbb{C} \vDash C$ if and only if there exists a resolution derivation of C from $\mathbb{C}$.

Proposition 2.5 (Implicational completeness of resolution). *Suppose $\mathbb{C}$ is a set of clauses and C is a clause, both over a set of variables of size n. Then $\mathbb{C} \vDash C$ if and only if there exists a resolution derivation of C from $\mathbb{C}$. Furthermore, if C can be derived from $\mathbb{C}$ then it can be derived in length at most $2^{n+1} - 1$ and variable space at most $n(n+2)$ simultaneously.*

Proof sketch. Suppose first that $C = 0$ is the contradictory empty clause. Build a search tree where all vertices on level i query the ith variable and where we go to the left, say, if the variable is false under a given truth value assignment α and to the right if the variable is true. As soon as some clause in $\mathbb{C}$ is falsified by the partial assignment defined by the path to a vertex, we make that vertex into a leaf labelled by that clause. This tree has height $h \leq n$ and hence size at most $2^{h+1} - 1$, and if we turn it upside down we can obtain a legal tree-like refutation (without weakening) of $\mathbb{C}$ in this length. This refutation can be carried out in clause space $h + 2$ and variable space upper-bounded by the clause space times the number of distinct variables, i.e., at most $n(n + 2)$. (We refer to, for instance, [BS02, ET01] for more details.)

If $C \neq 0$, apply the unique minimal restriction ρ falsifying C. Then $\mathbb{C}\!\upharpoonright_\rho \vDash C\!\upharpoonright_\rho = 0$, and we can construct a refutation of $\mathbb{C}\!\upharpoonright_\rho$ from a search tree of height $h < n$, since $\mathbb{C}\!\upharpoonright_\rho$ contains strictly fewer variables than $\mathbb{C}$. Removing the restriction ρ from this refutation, and adding at most one extra weakening step for every other derivation step (this is an example of where the weakening rule comes in handy), we get a derivation of C from $\mathbb{C}$. (See [BSW01] for a formal proof of this fact.) This derivation has length at most $2 \cdot (2^{h+1} - 1) < 2^{n+1} - 1$ and variable space at most $n(h + 2) < n(n + 2)$. □

2.3 Substitution Formulas

Throughout this paper, we will let f_d denote any (non-constant) Boolean function $f_d : \{0,1\}^d \mapsto \{0,1\}$ of arity d. We use the shorthand $\vec{x} = (x_1, \ldots, x_d)$, so that $f_d(\vec{x})$ is just an equivalent way of writing $f_d(x_1, \ldots, x_d)$. Every function $f_d(x_1, \ldots, x_d)$ is equivalent to a CNF formula over $x_1, \ldots, x_d$ with at most 2^d clauses. Fix a canonical way to represent functions as CNF formulas and let $Cl[f_d(\vec{x})]$ denote the canonical set of clauses representing f_d. Similarly, let $Cl[\neg f_d(\vec{x})]$ denote the clauses in the canonical representation of the negation of f. The following definition extends the notion of substitution to a CNF formula F. For notational convenience, we assume that F only has variables x, y, z, et cetera, without subscripts, so that $x_1, \ldots, x_d$, $y_1, \ldots, y_d$, $z_1, \ldots, z_d, \ldots$ are new variables not occurring in F. We will say that the variables $x_1, \ldots, x_d$, and any literals over these variables, all *belong* to the variable x.

[5] And indeed, for any reasonable proof complexity measure whatsoever, but we do not want to get too formal here by discussing what "reasonable" would mean in this context.

Definition 2.6 (Substitution formula). For a positive literal x and a non-constant Boolean function f_d, we define the f_d-*substitution* of x to be $x[f_d] = Cl[f_d(\vec{x})]$, i.e., the canonical representation of $f_d(x_1, \ldots, x_d)$ as a CNF formula. For a negative literal $\neg y$, the f_d-substitution is $\neg y[f_d] = Cl[\neg f_d(\vec{y})]$. The f_d-substitution of a clause $C = a_1 \vee \cdots \vee a_k$ is the CNF formula

$$C[f_d] = \bigwedge_{C_1 \in a_1[f_d]} \cdots \bigwedge_{C_k \in a_k[f_d]} \big(C_1 \vee \ldots \vee C_k\big) \tag{7}$$

and the f_d-substitution of a CNF formula F is $F[f_d] = \bigwedge_{C \in F} C[f_d]$.

As an example, for the clause $C = x \vee \overline{y}$ and for $f_2(x_1, x_2) = x_1 \oplus x_2$ being exclusive or, we get that

$$\begin{aligned} C[f_2] = \ & (x_1 \vee x_2 \vee y_1 \vee \overline{y}_2) \wedge (x_1 \vee x_2 \vee \overline{y}_1 \vee y_2) \\ & \wedge (\overline{x}_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2) \ . \end{aligned} \tag{8}$$

3 Main Results

In this section we state our main results. We start with the hierarchy theorem and the main theorem needed to prove it, the substitution space theorem. Then we discuss the main technical part of the paper: the size of minimally unsatisfiable k-DNF sets.

3.1 k-DNF Resolution Space Hierarchy and the Substitution Space Theorem

Our main theorem is the following.

Theorem 3.1 (k-DNF resolution space hierarchy). *For every $k \geq 1$ there exists an efficiently constructible family of formulas $\{F_n\}_{n=1}^{\infty}$ satisfying the following properties.*

1. *F_n is an unsatisfiable $(3(k+1))$-CNF formula with $O(n)$ variables and $O(n)$ clauses.*
2. *F_n can be refuted in $\mathfrak{R}(k+1)$ in length $O(n)$ and variable space $\mathrm{O}(1)$ simultaneously.*
3. *Every $\mathfrak{R}(k)$-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.

We want to stress that the upper bound on refutation space in $\mathfrak{R}(k+1)$ is stated in terms of the largest space measure—variable space—and hence holds also for formula space, whereas the lower bound on refutation space in $\mathfrak{R}(k)$ is stated using the smallest space measure, namely, formula space, and hence holds also for variable space.

The space hierarchy theorem follows from the next theorem describing how the space requirements of refuting a formula F in k-DNF resolution is affected by performing substitutions as in Definition 2.6. After presenting this "substitution space theorem,", we show how to derive the space hierarchy theorem from it. To state the theorem we need the following definition.

Definition 3.2 (Non-authoritarian functions). We say that a Boolean function f over variables $X = \{x_1, \ldots, x_d\}$ is k-*non-authoritarian* if no restriction to X of size k can fix the value of f. In other words, for every restriction ρ to X with $|\rho| \leq k$ there exist two assignments $\alpha_0, \alpha_1 \supset \rho$ such that $f(\alpha_0) = 0$ and $f(\alpha_1) = 1$.

Notice that a function on d variables can be k-non-authoritarian only if $k < d$. The XOR function $\oplus$ on d variables is $(d-1)$-non-authoritarian and the majority function on $2d'+1$ variables is d'-non-authoritarian.

The substitution space theorem for $\mathfrak{R}(k)$ tells us that for 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. To parse the bounds below more easily, the reader might be helped by thinking of c, d, and k as constants (which they will be in our application of the theorem).

Theorem 3.3 (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. Then the following two properties hold for the substitution formula $F[f_d]$:*

1. *If F can be refuted in resolution in length L and variable space S simultaneously, then $F[f_d]$ can be refuted in $\mathfrak{R}(d)$ in length $L \cdot d^{4^{cd}} \cdot 4^{cd}$ and variable space $S \cdot d2^d + (cd+2)^3 \cdot 4^{cd} + O(1)$ simultaneously.*

2. *If f_d is k-non-authoritarian and $F[f_d]$ can be refuted by a $\mathfrak{R}(k)$-refutation that requires space S' and makes L' axiom downloads, then F can be refuted by a resolution refutation that requires variable support size at most $(2S'k)^{k+1} \cdot 4^{k^2 d}$ and makes at most L' axiom downloads.*

Assuming this theorem, we can establish the k-DNF resolution space hierarchy.

Proof of Theorem 3.1. The first author described in [BS02, Theorems 3.1 and 3.2] a family of efficiently constructible 3-CNF formulas $\{F'_n\}_{n=1}^{\infty}$ satisfying:

- F'_n can be refuted in resolution in length $O(n)$ and variable space $\mathrm{O}(1)$ simultaneously.
- Every resolution refutation of F'_n has variable support size[6] $\Omega(n/\log n)$.

(The family $\{F'_n\}_{n=1}^{\infty}$ consists of so-called pebbling contradictions over directed acyclic graphs G_n with n vertices that have a black-white pebbling price of $\Omega(n/\log n)$. We refer the interested reader to the second author's PhD thesis [Nor08] and upcoming survey [Nor09] for further information about pebble games and their applications to proof complexity.)

The family $\{F_n\}_{n=1}^{\infty}$ is obtained by substituting an arbitrary non-authoritarian Boolean function f_{k+1} of arity $k+1$, for instance XOR over $k+1$ variables, into F'_n, i.e., by setting $F_n = F'_n[f_{k+1}]$ for all $n \in \mathbb{N}^+$. By construction F_n satisfies part 1 of Theorem 3.1. To obtain the remaining two parts of the theorem, we apply Theorem 3.3 to F'_n. Using part 1 of this theorem and noticing that in our case $d = k+1$ and $c = 3(k+1)$ are constants, we conclude that F_n can be refuted in $(k+1)$-DNF resolution in linear length and constant space simultaneously, thus yielding part 2 of Theorem 3.1. To obtain part 3 we use the lower bound of $\Omega(n/\log n)$ on the variable support size of F'_n and combine it with part 2 of Theorem 3.3. This completes the proof of Theorem 3.1. □

3.2 Minimally unsatisfiable k-DNF formula sets

The proof of the first part of Theorem 3.3 is fairly straightforward and resembles our proof of the substitution theorem for the standard resolution proof system in [BSN09]. 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 in this section.

[6] The exact statement in [BS02] says that the variable space of F'_n is at least $\Omega(n/\log n)$. However, the proof given there actually shows a lower bound on the variable support size.

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 of this notion is to define $\mathbb{D}$ to be minimally unsatisfiable if it is unsatisfiable but all proper subsets of it are satisfiable. The following example shows why this approach is problematic.

$$\{x, \big((\overline{x} \wedge y_1) \vee (\overline{x} \wedge y_2) \vee (\overline{x} \wedge y_3) \vee \cdots \vee (\overline{x} \wedge y_n)\big)\} \tag{9}$$

This set, 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 3.4 (Minimal implication and minimally unsatisfiable k-DNF sets). Let $\mathbb{D}$ be a DNF set and G be a formula. We say $\mathbb{D}$ *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, hence can be removed from F.

The following theorem is our extension of Theorem 1.1.

Theorem 3.5 (Small-size minimally unsatisfiable k-DNF sets have few variables). *Suppose that $\mathbb{D}$ is a minimally unsatisfiable k-DNF set. Then the number of variables in $\mathbb{D}$ is at most $|\mathit{Vars}(\mathbb{D})| \leq (k \cdot |\mathbb{D}|)^{k+1}$.*

We want to point out that in contrast to Theorem 1.1, which is exactly tight, there is no matching lower bound on the number of variables in Theorem 3.5. And indeed, we see no particular reason to believe that this theorem should be tight. We note that the best explicit construction of a minimally unsatisfiable k-DNF set that we are currently able to obtain have 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 1.1.

Lemma 3.6 (Explicit construction of minimally unsatisfiable k-DNF set). *There are minimally unsatisfiable k-DNF sets $\mathbb{D}$ with $|\mathit{Vars}(\mathbb{D})| \geq k^2(|\mathbb{D}| - 1)$.*

Proof. Consider any minimally unsatisfiable CNF formula consisting of $n + 1$ clauses over n variables (for instance, the one in (2)). Substitute every variable x_i with

$$\big(x_i^1 \wedge x_i^2 \wedge \cdots \wedge x_i^k\big) \vee \big(x_i^{k+1} \wedge x_i^{k+2} \wedge \cdots \wedge x_i^{2k}\big) \vee \cdots \vee \big(x_i^{k^2-k+1} \wedge x_i^{k^2-k+2} \wedge \cdots \wedge x_i^{k^2}\big) \tag{10}$$

and expand every clause to a k-DNF formula. Note that this is possible since the negation of (10) that we need to substitute for $\neg x_i$ can also be expressed as a k-DNF formula

$$\bigvee_{(j_1,\ldots,j_k)\in[1,k]\times\ldots\times[(k^2-k+1,k^2]} \big(\neg x_i^{j_1} \wedge \cdots \wedge \neg x_i^{j_k}\big) \ . \tag{11}$$

It is straightforward to verify that the result is a minimally unsatisfiable k-DNF set in the sense of Definition 3.4, and this set has $n + 1$ formulas over $k^2 n$ variables. ☐

We end this section by remarking that the precise statement required to prove the second part of Theorem 3.3 (found in Lemma 5.4) is somewhat more involved than Theorem 3.5. However, the proof of Lemma 5.4 follows closely the proof of Theorem 3.5. Both of these results are proved in sequence in Section 5.

4 $\mathfrak{R}(k)$-refutations of $F[f]$ Translate into Resolution Refutations of F

To prove part 2 of Theorem 3.3, we need to show how to convert a $\mathfrak{R}(k)$-refutation π_f of $F[f_d]$ into a resolution refutation π of F such that the variable support size of π is bounded by the space of π_f, raised to the power of $k+1$. The proof has two main parts. In Lemma 4.2 we claim that each k-DNF set $\mathbb{D} \in \pi_f$ can be "projected" onto a set of clauses over $\mathit{Vars}(F)$, such that the sequence of projected clause sets forms the "backbone" of a resolution refutation of F. By this we mean that the backbone can be completed to a standard resolution refutation of F without (essentially) increasing the variable support size. Then, in Lemma 4.3, which forms the second and main part of the proof, we show that if $\mathbb{C}$ is a set of clauses projected by a k-DNF set $\mathbb{D}$, the variable support size of $\mathbb{C}$ is at most $|\mathbb{D}|^{k+1}$. Combining these lemmas proves part 2 of Theorem 3.3.

This section is organized as follows. We start by formally defining the set of clauses "projected" by a k-DNF set. Then we state the two main lemmas regarding projected proofs. After completing the proof of part 2 of Theorem 3.3, we attend to the proofs of the lemmas.

The clauses projected by a k-DNF set $\mathbb{D}$ are those clauses that are *precisely implied* by $\mathbb{D}$ according to the following definition.

Definition 4.1 (Precise implication and projected clauses). Let F be a CNF formula and f_d a non-constant Boolean function, and suppose that $\mathbb{D}$ is a k-DNF set derived from $F[f_d]$ and that P and N are (disjoint) subsets of variables of F. If

$$\mathbb{D} \vDash \bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y}) \tag{12a}$$

but for all strict subsets $P' \subsetneqq P$, and $N' \subsetneqq N$ it holds that

$$\mathbb{D} \nvDash \bigvee_{x \in P'} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y}) \ , \text{ and} \tag{12b}$$

$$\mathbb{D} \nvDash \bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N'} \neg f_d(\vec{y}) \ , \tag{12c}$$

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* and write

$$\mathbb{D} \rhd \bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y}) \ . \tag{13}$$

Letting $C = C^+ \vee C^-$ be the clause defined by $C^+ = \bigvee_{x \in P} x$ and $C^- = \bigvee_{y \in n} \neg y$, we say that $\mathbb{D}$ *projects* the clause C if (13) holds. Finally, we let

$$\mathit{proj}_F(\mathbb{D}) = \left\{ C \,\middle|\, \mathbb{D} \rhd \textstyle\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y}) \right\} \tag{14}$$

denote the set of all clauses that $\mathbb{D}$ projects on F.

Informally, the next lemma states that the projection of a k-DNF resolution refutation of $F[f_d]$ is essentially a refutation of the original formula F in standard resolution. And more importantly for our purposes, small $\mathfrak{R}(k)$-refutation space implies the projected resolution refutation has small variable support size. The proof of this lemma appears in Section 6.

Lemma 4.2 (Basic properties of projected proof). *Let $k \geq 1$. Suppose that $\pi_f = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ is a $\mathfrak{R}(k)$-refutation of $F[f_d]$ for some arbitrary unsatisfiable CNF formula F and some arbitrary non-constant function f_d. Then the sets of projected clauses $\{proj_F(\mathbb{D}_0), \ldots, proj_F(\mathbb{D}_\tau)\}$ form the "backbone" of a resolution refutation π of F in the sense that:*

- $proj_F(\mathbb{D}_0) = \emptyset$.
- $proj_F(\mathbb{D}_\tau) = \{0\}$.
- *The only time π performs a download of some axiom C in F is when π_f downloads some axiom $D \in C[f_d]$ in $F[f_d]$.*
- *All transitions from $proj_F(\mathbb{D}_{t-1})$ to $proj_F(\mathbb{D}_t)$ for $t \in [\tau]$ can be accomplished by axiom downloads from F, resolution inferences, erasures, and possibly resolution weakening steps in such a way that the variable support size in π during these intermediate derivation steps never exceeds $2 \cdot \max_{\mathbb{D} \in \pi_f}\{SuppSize(proj_F(\mathbb{D}))\}$.*

The following statement is the main technical part of our argument. Its proof is deferred to the next section.

Lemma 4.3 (Main lemma—lower bound on space of projected proof). *Suppose that F is a CNF formula and f_d is a k-non-authoritarian function of arity $d > k$ and $\mathbb{D}$ is a k-DNF set over $Vars(F[f_d])$. Then it holds that*

$$SuppSize(proj_F(\mathbb{D})) \leq 4^{k^2 d} \cdot (k \cdot Sp(\mathbb{D}))^{k+1} \quad .$$

Given the two lemmas above we proceed to prove the second part of the substitution space theorem.

Proof of part 2 of Theorem 3.3. Let F be an unsatisfiable c-CNF formula and f_d a non-constant Boolean function of arity $d > k \geq 1$. Let π_f be a $\mathfrak{R}(k)$-refutation of $F[f_d]$ that requires space S' and makes L' axiom downloads. By Lemma 4.2, the sequence of sets of clauses $\pi' = \{proj_F(\mathbb{D}_0), \ldots, proj_F(\mathbb{D}_\tau)\}$ can be extended to a resolution refutation π of F such that the number of axiom downloads in π is L' and the variable support size of π is at most $2S'$. Additionally, Lemma 4.3 implies that the maximal support size of π' is bounded by $(2kS')^{k+1} \cdot 4^{k^2 d}$ and this number is also an upper bound on the maximal support size of π as well. This completes the proof of part 2 of Theorem 3.3. ☐

5 On the Size of Minimally Implicating k-DNF Sets

In this section we prove Lemma 4.3, which bounds the number of variables appearing in a k-DNF set that minimally implies a formula. We first deal with the special case of a minimally unsatisfiable set in Section 5.1. The actual result needed to prove the substitution space theorem follows the outline of this simpler case and appears in Section 5.2.

5.1 Warmup—on the Size of Minimally Unsatisfiable k-DNF Sets

In this subsection we prove Theorem 3.5. The following simple but important lemma will be used both in the proof of Theorem 3.5 and of Lemma 4.3. We state it in the more general form needed to prove the latter result. (For proving Theorem 3.5 it suffices to restrict our attention to *unsatisfiable* formulas G.)

Lemma 5.1. *Suppose that $\mathbb{D}$ is a k-DNF set that minimally implies a formula G. Then for every literal a appearing in any term T in a k-DNF formula $D \in \mathbb{D}$ there exists a restriction ρ to $\mathit{Vars}(\mathbb{D})$ satisfying*

- $|\rho| \leq k|\mathbb{D}|$.
- $D'{\restriction_\rho} = 1$ *for all* $D' \in \mathbb{D} \setminus \{D\}$.
- $(T \setminus \{a\}){\restriction_\rho} = 1$.
- $G{\restriction_\rho} \neq 1$.

The point here is that, intuitively speaking, the restriction ρ is very nearly satisfying the k-DNF set $\mathbb{D}$ (except for a single literal in a single term) but still has not fixed the formula G implied by $\mathbb{D}$ to true. Also, ρ assigns values to comparatively few variables.

Proof of Lemma 5.1. By Definition 3.4, there exists an assignment α to $\mathit{Vars}(\mathbb{D})$ such that

- $D'(\alpha) = 1$ for all $D' \in \mathbb{D}' \setminus \{D\}$.
- $(T \setminus \{a\})(\alpha) = 1$.
- $G(\alpha) = 0$.

Let ρ be a restriction of minimal size that agrees with α and satisfies the second and third bullet in the statement of the lemma. Such a restriction can be found by selecting one term T' satisfied by α in each $D' \in \mathbb{D}' \setminus D$ and setting ρ to agree with α on $\bigcup_j \mathit{Vars}(T_j) \cup \mathit{Vars}(T \setminus \{a\})$ and be unfixed elsewhere. Since $|T'| \leq k$ we see ρ has size $\leq k|\mathbb{D}'|$. The last bullet stated above holds because $G(\alpha) = 0$ and ρ agrees with α on all variables fixed by ρ. ☐

We now bound the number of variables appearing in a minimally unsatisfiable k-DNF set.

Proof of Theorem 3.5. Let $\mathbb{D} = \{D_1, \ldots, D_m\}$ be a k-DNF formula set with $m = |\mathbb{D}|$. For S a set of literals, let $D_i(S)$ be the set of terms in D_i that contain S (recall we identify a term with the set of literals appearing in it). Formally,

$$D_i(S) = \{T \in D_i : T \supseteq S\} \ . \tag{15}$$

Let $\mathit{Vars}(D_i(S))$ denote the set of variables appearing in the set of terms $D_i(S)$. Our theorem follows from the next claim.

Claim 5.2. If S is a set of literals and $|S| = k - r$ then $|\mathit{Vars}(D_i(S))| \leq k \cdot (km)^r$.

Before proving the claim let us complete the proof of the theorem. Take $S = \emptyset$ for which we get $r = k$ and notice that $D_i(\emptyset) = D_i$. Claim 5.2 gives

$$|\mathit{Vars}(D_i)| = |\mathit{Vars}(D_i(\emptyset))| \leq k(km)^k \tag{16}$$

and summing over all all m formulas in the set we get

$$|\mathit{Vars}(\mathbb{D})| \leq \textstyle\sum_{i=1}^{m} |\mathit{Vars}(D_i)| \leq m \cdot k(km)^k = (km)^{k+1} = (k|\mathbb{D}|)^{k+1} \tag{17}$$

which concludes the proof. ☐

Proof of Claim 5.2. By induction on $r \geq 0$. For the base case of $r = 0$ notice $|S| = k$ so there can be at most one term in D_i that contains all literals in S implying $|\mathit{Vars}(D_i(S))|$ is either 0 or k.

For the inductive step we may assume the existence of some term $T \in D_i$ that strictly contains S, because otherwise S appears at most once as a term in D_i and the claim holds as in the base case. Assuming $T \supsetneq S$, let a be a literal in $T \setminus S$. Lemma 5.1 guarantees the existence of a restriction ρ of size at most km such that $D_j\restriction_\rho = 1$ for all $j \in [m], j \neq i$ and $(T \setminus \{a\})\restriction_\rho \neq 0$. By the unsatisfiability of $\mathbb{D}$ we conclude ρ falsifies every term $T' \in D_i$ for which $T' \supsetneq S$. Since $(T \setminus \{a\})\restriction_\rho = 1$ and $(T \setminus \{a\}) \supseteq S$ we conclude that every term in D_i that contains S must also contain a literal set to false by ρ, because otherwise ρ could be extended to an assignment satisfying $\mathbb{D}$. Recall that $\neg\rho$ is the set of literals set to false by ρ. We have just shown that

$$D_i(S) = \bigcup_{a' \in \neg\rho} D_i(S \cup \{a'\}) \ . \tag{18}$$

So to bound $\mathit{Vars}(D_i(S))$ we need only bound $\mathit{Vars}(D_i(S \cup \{a'\}))$ for all $a' \in \neg\rho$. We use the inductive hypothesis. Notice $(\neg\rho) \cap S = \emptyset$ because ρ satisfies S. Thus, for $a' \in \neg\rho$ we have $|S \cup \{a'\}| = k - (r-1)$. Summing over all $a' \in \neg\rho$ and recalling $|\neg\rho| = |\rho| \leq km$, we apply the inductive hypothesis to $S \cup \{a'\}$ to conclude from (18) that

$$|\mathit{Vars}(D_i(S))| \leq \sum_{a' \in \neg\rho} |\mathit{Vars}(D_i(S \cup \{a'\}))| \leq |\neg\rho| \cdot k(km)^{r-1} \leq k(km)^r \tag{19}$$

as claimed. □

5.2 Upper-bounding the Space of Projections—Proof of Lemma 4.3

To prove Lemma 4.3 we need to address two issues that did not appear in the previous subsection. First, our starting point is a k-DNF set $\mathbb{D}$ that is satisfiable and implies a set of projected clauses. We deal with this by constructing a formula (denoted G' later on) that is the conjunction of all clauses projected by $\mathbb{D}$. The second issue, which is more subtle, is that $\mathbb{D}$ is a set of formulas defined over $\mathit{Vars}(F[f_d])$ whereas the clauses projected by $\mathbb{D}$ are over the different variable set $\mathit{Vars}(F)$. The following definition will be used to connect the two sets of variables and is crucial to our proof.

Definition 5.3 (Shadow). For a a literal over a variable $y \in \mathit{Vars}(F[f_d])$ let the *shadow* of a, denoted $\mathbf{V}(a)$, be the variable $x \in \mathit{Vars}(F)$ to which a belongs, i.e., the shadow of y is the variable x such that $y \in \mathit{Vars}(x[f_d])$. For T a set of literals (which will later on be identified with a term or a restriction) let its shadow be $\mathbf{V}(T) = \bigcup_{a \in T} \mathbf{V}(a)$ and for D a set of terms we define its shadow as $\mathbf{V}(D) = \bigcup_{T \in D} \mathbf{V}(T)$.

The following sublemma, which will be proved later on, is the analog of Claim 5.2, accounting for the needed modifications which were discussed in the beginning of this section. The claim in this sublemma is also the central point in our proof of Lemma 4.3. We now state the sublemma and promptly use it to complete the proof of Lemma 4.3.

Lemma 5.4. *Suppose $\mathbb{D} = \{D_1, \ldots, D_m\}$ is a k-DNF set over $\mathit{Vars}(F[f_d])$ and G is a CNF formula over $\mathit{Vars}(F)$ such that $\mathbb{D}$ minimally implies the substituted formula $G' = G[f_d]$. Suppose furthermore that $S \subset \mathit{Vars}(F)$ and $|S| = k - r$ for $r \geq 0$. Then, letting $D_i(S) = \{T \in D_i | \mathbf{V}(T) \supseteq S\}$ denote the set of terms in D_i whose shadow contains S, we have*

$$|\mathbf{V}(D_i(S))| \leq k \cdot \left(4^{kd} \cdot k|\mathbb{D}|\right)^r .$$

Proof of Lemma 4.3. Let $\mathbb{D} = \{D_1, \ldots, D_m\}$ and $G' = \bigwedge_{C \in \mathit{proj}_F(\mathbb{D})} C[f_d]$. Notice that by Definition 4.1, G' is of the form $G' = G[f_d]$ for some CNF formula G over $\mathit{Vars}(F)$ so G' conforms to the assumptions of Lemma 5.4.

First we argue that we may assume without loss of generality that $\mathbb{D}$ minimally implies G'. If this is not the case, there must exist a term T appearing in $D_i \in \mathbb{D}$ and a proper subterm $T' \subsetneq T$ such that replacing T by T' and calling the replaced k-DNF set by $\mathbb{D}'$, we still have $\mathbb{D}' \vDash G'$. In this case replace $\mathbb{D}$ with $\mathbb{D}'$ and repeat the process. Notice that repeating the process does not increase the size of $\mathbb{D}$ (in fact, the size can shrink if some k-DNF formula includes an empty term). Since each repetition of this process strictly shrinks the number of literals in $\mathbb{D}$ (counted with repetitions), we see it must terminate. Upon termination the remaining k-DNF set, denoted $\hat{\mathbb{D}}$, which is of size at most m, minimally implies G'.

Our next observation is that for every variable x appearing in G there must exist a literal a belonging to it that appears in $\hat{\mathbb{D}}$, implying

$$\mathit{SuppSize}(G) = |\mathit{Vars}(G)| \leq |\mathbf{V}(\hat{\mathbb{D}})| \ . \tag{20}$$

To see this, argue by way of contradiction. Let $C = C' \vee x$ be a clause appearing in G and assume for simplicity that x is a positive literal (the case of a negative literal is identical). Conditions (12a) and (12b) of Definition 4.1 imply that there exists an assignment α to $\mathit{Vars}\big(F[f_d]\big)$ such that $\alpha(\mathbb{D}) = \alpha(x[f_d]) = 1$ but $\alpha(C'[f_d]) = 0$. By construction, $\mathbb{D} \vDash \hat{\mathbb{D}}$ so $\alpha(\hat{\mathbb{D}}) = 1$ as well. By assumption, no variable belonging to x appears in $\hat{\mathbb{D}}$, so by changing the value of α on $\mathit{Vars}\big(x[f_d]\big)$ as to falsify $x[f_d]$ we reach an assignment that satisfies $\hat{\mathbb{D}}$ but falsifies $G[f_d]$, contradiction.

Having established (20), we bound $|\mathbf{V}(D)|$ for $D \in \hat{\mathbb{D}}$ with the use of Lemma 5.4 and get

$$|\mathbf{V}(D)| = |\mathbf{V}(D(\emptyset))| \leq k \cdot \left(4^{kd} \cdot k|\hat{\mathbb{D}}|\right)^k . \tag{21}$$

Summing over all $D \in \mathbb{D}$ and recalling $|\hat{\mathbb{D}}| \leq |\mathbb{D}|$ gives

$$|\mathbf{V}(\hat{\mathbb{D}})| \leq \sum_{D \in \hat{\mathbb{D}}} |\mathbf{V}(D)| \leq |\hat{\mathbb{D}}| \cdot k \cdot \left(4^{kd} \cdot k|\hat{\mathbb{D}}|\right)^k \leq 4^{k^2 d} \cdot (k|\mathbb{D}|)^{k+1} \tag{22}$$

and this, together with (20), completes the proof of Lemma 4.3. □

We end this section with a proof of Lemma 5.4.

Proof of Lemma 5.4. By induction on $r \geq 0$. For the base case of $r = 0$ we have $|S| = k$. Since D_i is a k-DNF formula then any term T for which $\mathbf{V}(T) \supseteq S$ must have $\mathbf{V}(T) = S$. Thus, $|\mathbf{V}(D_i(S))| = k$ and the inequality claimed in the lemma holds.

For the inductive case of $r > 0$, let $\bar{S}$ denote the set of literals that belong to S, and let $\mathrm{terms}(S)$ denote the set of terms over $\bar{S}$. We bound the number of terms by $|\mathrm{terms}(S)| = 2^{2|\mathit{Vars}(\bar{S})|} \leq 4^{kd}$ because each term is a set of literals coming from a set of literals of size $2|\mathit{Vars}\big(\bar{S}\big)|$. Partition the terms in $D_i(S)$ according to their intersection with $\bar{S}$. Formally, for every $s \in \mathrm{terms}(S)$ let

$$D_i(s) = \{T \in D_i(S) | T \cap \bar{S} = s\} \ . \tag{23}$$

We have partitioned $D_i(S)$ into 4^{kd} partitions so to prove the claim in the lemma it is sufficient to show for each partition that

$$|\mathbf{V}(D_i(s))| \leq km \left(k \cdot \left(4^{kd} km\right)^{r-1}\right) \ . \tag{24}$$

Consider one term $s \in \mathrm{terms}(S)$. If $\mathbf{V}(D_i(s)) = S$ then clearly (24) holds so we assume $\mathbf{V}(D_i(s)) \supsetneq S$. In this case there exists $T \in D_i$ such that $\mathbf{V}(T) \supsetneq S$ which implies the existence of a literal $a \in T \setminus \bar{S}$. Let ρ be a restriction satisfying the properties of Lemma 5.1 with respect to a, T, D_i and G'.

Proposition 5.5. *Every term T' appearing in $D_i(s)$ must include a literal $a \notin \bar{S}$ whose shadow belongs to the shadow of ρ as well. Formally, $(\mathbf{V}(T') \setminus S) \cap (\mathbf{V}(\rho) \setminus S) \neq \emptyset$.*

Proof. By way of contradiction. Assume T' falsifies the proposition. By assumption T' has the same set of literals as T within $\bar{S}$ and the third property of ρ listed in Lemma 5.1 implies ρ satisfies all literals of T' inside $\bar{S}$. Assuming that the intersection in the statement of the proposition is empty, we can extend ρ to a restriction ρ' that satisfies T' by setting at most k "new" variables on top of those set by ρ. The crucial observation is that none of the "new" variables set by ρ' have their shadow in $\mathbf{V}(\rho)$. More to the point, suppose x_i is a "new" variable whose value is set by ρ' but is not set by ρ. Let x denote the shadow of x_i and let $\vec{x} = \{x_1, \ldots, x_d\}$ be the set of variables whose shadow is x. Our crucial observation, restated in different words, is that ρ does not set the value of *any* variable in $\vec{x}$. This is where the k-non-authoritarianism of f_d comes into play, because it implies that ρ' cannot fix the value of $f_d(\vec{x})$ because ρ' sets at most k variables in $\vec{x}$. But this means that we can extend ρ' so that $f_d(\vec{x})$ will obtain any truth value we find fit. We conclude that the fourth property listed in Lemma 5.1 holds for ρ' as well as for ρ. This property implies that ρ' can be extended to an assignment α' such that $G'(\alpha') = 0$. So α' is an assignment that satisfies $\mathbb{D}$ but falsifies G'. We have reached a contradiction, and the proposition follows. ☐

We continue with the proof of the inequality (24). The second property of Lemma 5.1 implies that $|\mathbf{V}(\rho)| \leq km$. Thus, Proposition 5.5 shows that there exists a set $V_s \subseteq \mathit{Vars}(F) \setminus S$ of size at most km such that

$$\mathbf{V}(D_i(s)) \subseteq \bigcup_{v \in V_s} \mathbf{V}(D_i(S \cup \{v\})) \ . \tag{25}$$

Since $v \notin S$ we have $|S \cup \{v\}| = k - (r-1)$ so we may apply the inductive hypothesis of the inequality in Lemma 5.4 to $S \cup \{v\}$ which gives

$$|\mathbf{V}(D_i(s))| \leq \sum_{v \in V_s} |\mathbf{V}(D_i(S \cup \{v\}))| \leq km \left(k \cdot \left(4^{kd} \cdot km\right)^{r-1}\right) \ . \tag{26}$$

We have shown that the inequality (24) holds for all $s \in \mathrm{terms}(S)$. Summing over all terms, there are at most 4^{kd} of them, completes the proof of Lemma 5.4. ☐

6 Projected $\mathfrak{R}(k)$-refutations Are (Almost) Resolution Refutations

This section contains the proof of Lemma 4.2. We establish this lemma in very much the same way as for [BSN09, Theorem 4.4], but there is a subtle difference between the two proofs due to the fact that our definition of precise implication (Definition 4.1) is somewhat different than what is used there (cf. [BSN09, Definition 4.2]). Definition 4.2 in [BSN09] appears to be "the right one" and yields tighter results for standard resolution, but for technical reasons we are forced to relax that definition a bit in order to obtain the results for k-DNF resolution in the current paper.

We first fix some notation. Let us use the convention that $\mathbb{D}$ and D denote k-DNF sets and k-DNF formulas derived from $F[f_d]$ while $\mathbb{C}$ and C denote clause sets and clauses derived from F. Let us also overload the notation and write $\mathbb{D} \vDash C$, $\mathbb{D} \nvDash C$, and $\mathbb{D} \rhd C$ for $C = C^+ \vee C^-$ when the corresponding implications hold or do not hold for $\mathbb{D}$ with respect to $\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})$. Finally, let $\mathbb{C}_t$ be a shorthand for $\mathit{proj}_F(\mathbb{D}_t)$.

Suppose now that $\pi_f = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ is a k-DNF resolution refutation of $F[f_d]$ for some arbitrary unsatisfiable CNF formula F and some arbitrary non-constant function f_d.

The first two bullets in Lemma 4.2 are immediate. For $\mathbb{D}_0 = \emptyset$ we have $\mathbb{C}_0 = \mathit{proj}_F(\mathbb{D}_0) = \emptyset$, and it is easy to verify that $\mathbb{D}_\tau = \{0\}$ yields $\mathbb{C}_\tau = \mathit{proj}_F(\mathbb{D}_\tau) = \{0\}$. We note, however, that the empty clause will have appeared in $\mathbb{C}_t = \mathit{proj}_F(\mathbb{D}_t)$ earlier, namely for the first t such that $\mathbb{D}_t$ is contradictory.

The hard part is to show that all transitions from $\mathbb{C}_{t-1} = \mathit{proj}_F(\mathbb{D}_{t-1})$ to $\mathbb{C}_t = \mathit{proj}_F(\mathbb{D}_t)$ can be performed in such a way that the variable support size in our refutation under construction $\pi : F \vdash 0$ never exceeds $\mathit{SuppSize}(\mathbb{C}_{t-1}) + \mathit{SuppSize}(\mathbb{C}_t) \leq 2 \cdot \max_{s \in [\tau]} \{\mathit{SuppSize}(\mathbb{C}_s)\}$ during the intermediate derivation steps needed in π. The proof is by a case analysis of the derivation steps. Before plunging into the proof, let us make a simple but useful observation.

Observation 6.1. *Using the above notation, if $\mathbb{D}_t \vDash C$ then $C = C^+ \vee C^-$ is derivable from $\mathbb{C}_t = \mathit{proj}_F(\mathbb{D}_t)$ by weakening.*

Proof. Pick $C_1^+ \subseteq C^+$, and $C_2^- \subseteq C^-$ minimal so that $\mathbb{D} \vDash C_1^+ \vee C_2^-$ still holds. Then by definition $\mathbb{D} \rhd C_1^+ \vee C_2^-$ so $C_1^+ \vee C_2^- \in \mathbb{C}_t$ and $C \supseteq C_1^+ \vee C_2^-$ can be derived from $\mathbb{C}_t$ by weakening as claimed. ☐

Consider now the rule applied in π_f at time t to get from $\mathbb{D}_{t-1}$ to $\mathbb{D}_t$. We analyze the three possible cases—inference, erasure and axiom download—in this order.

Inference Note that obviously $\mathbb{D}_{t-1} \vDash \mathbb{D}_t$ since all inference rules are sound. Moreover, since $\mathbb{D}_t \supseteq \mathbb{D}_{t-1}$ we have $\mathbb{D}_t \vDash \mathbb{D}_{t-1}$. It follows from Definition 4.1 the set of projected clauses does not change, i.e., $\mathbb{C}_{t-1} = \mathbb{C}_t$, and nothing needs to be done.

Erasure If $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ is a new projected clause appearing at time t as a result of an erasure $\mathbb{D}_t = \mathbb{D}_{t-1} \setminus \{D\}$, it clearly holds that $\mathbb{D}_{t-1} \vDash C$. Hence, all such clauses $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived by weakening from $\mathbb{C}_{t-1}$ by Observation 6.1, after which all clauses in $\mathbb{C}_{t-1} \setminus \mathbb{C}_t$ can be erased. During these intermediate steps the support size is upper-bounded by $\mathit{SuppSize}(\mathbb{C}_{t-1} \cup \mathbb{C}_t) \leq \mathit{SuppSize}(\mathbb{C}_{t-1}) + \mathit{SuppSize}(\mathbb{C}_t)$.

Axiom download This is the place in the case analysis where we need to do some serious work. Suppose that $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{D\}$ for some axiom clause $D \in A[f_d]$, where A in turn is an axiom of F. If $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ is a new projected clause then we must have $\mathbb{D}_{t-1} \nvDash C$ and $\mathbb{D}_{t-1} \cup \{D\} \rhd C$.

We want to show that all such clauses C can be derived from $\mathbb{C}_{t-1} = \mathit{proj}_F(\mathbb{D}_{t-1})$ by downloading $A \in F$, making inferences, and then possibly erasing A, and that this can be done without the variable support size exceeding $\mathit{SuppSize}(\mathbb{C}_{t-1}) + \mathit{SuppSize}(\mathbb{C}_t)$. The key to our proof is the next lemma.

Lemma 6.2. *Let $\mathbb{D}$ be a k-DNF set derived from $D \in F[f_d]$, $D \in A[f_d]$ be an axiom clause of $F[f_d]$, and C be a clause over $\mathit{Vars}(F)$. If $\mathbb{D}$, D, and C are such that $\mathbb{D} \cup \{D\} \rhd C$ but $\mathbb{D} \nvDash C$. Then if $A = a_1 \vee \cdots \vee a_k$, for every $a_i \in A \setminus C$ there is a subclause $C^i \subseteq C$ such that $\mathbb{D} \rhd C^i \vee \overline{a}_i$. That is, all clauses $C \vee \overline{a}_i$ for $a_i \in A \setminus C$ can be derived from $\mathbb{C} = \mathit{proj}_F(\mathbb{D})$ by weakening.*

Proof. Consider any assignment α such that $\mathbb{D}(\alpha) = 1$ but $\left[\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})\right](\alpha) = 0$. Such an assignment exists since $\mathbb{D} \nvDash C$ by assumption. Also, since by assumption $\mathbb{D} \cup \{D\} \rhd C$ we must have $D(\alpha) = 0$. If $A = a_1 \vee \cdots \vee a_s$, we can write $D \in A[f_d]$ on the form $D = D_1 \vee \cdots \vee D_s$ for $D_i \in a_i[f_d]$. Fix any $a \in A$ and suppose for the moment that $a = x$ is a positive literal. Then $D_i(\alpha) = 0$ implies that $\left[f_d(\vec{x})\right](\alpha) = 0$ which means that $\left[\neg f_d(\vec{x})\right](\alpha) = 1$. Since exactly the same argument holds if $a = \overline{y}$ is a negative literal, we conclude that

$$\mathbb{D} \vDash \bigvee_{x \in (C \vee \overline{a}_i)^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in (C \vee \overline{a}_i)^-} \neg f_d(\vec{y}) \tag{27}$$

or, rewriting (27) using our overloaded notation, that

$$\mathbb{D} \vDash C \vee \overline{a}_i \ . \tag{28}$$

If $a_i \in C$, the clause $C \vee \overline{a}_i$ is trivially true and thus uninteresting, but otherwise we pick $C^i \subseteq C$ minimal such that (28) still holds (and notice that since $\mathbb{D} \nvDash C$, the literal $\overline{a}_i$ cannot be dropped from the implication). Then by Definition 4.1 we have $\mathbb{D} \rhd C^i \vee \overline{a}_i$ as claimed. □

We remark that Lemma 6.2 tells us that every $x \in \mathit{Vars}(A) \setminus \mathit{Vars}(C)$ appears in some clause at time $t-1$, namely, in the clause $C^i \vee \overline{a}_i$ found in the proof above. Since in addition obviously $\mathit{Vars}(A) \cap \mathit{Vars}(C) \subseteq \mathit{Vars}(\mathbb{C}_t)$ this means that if we download $A \in F$ in our refutation $\pi : F \vdash 0$ under construction, we have $\mathit{Vars}(A) \subseteq \mathit{Vars}(\mathbb{C}_{t-1}) \cup \mathit{Vars}(\mathbb{C}_t)$ and hence $\mathit{SuppSize}(\mathbb{C}_{t-1} \cup \{A\}) \leq \mathit{SuppSize}(\mathbb{C}_{t-1}) + \mathit{SuppSize}(\mathbb{C}_t)$.

Thus, we can download $A \in F$, and then possibly erase this clause again at the end of our intermediate resolution derivation to get from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$, without the variable support size ever exceeding $\mathit{SuppSize}(\mathbb{C}_{t-1}) + \mathit{SuppSize}(\mathbb{C}_t)$. Let us now argue that all new clauses $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived from $\mathbb{C}_{t-1} \cup \{A\}$.

If $A \setminus C = \emptyset$, then the weakening rule applied on A is enough. Suppose therefore that this is not the case and let $A' = A \setminus C = \bigvee_{a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)} a$. Appealing to Lemma 6.2, we know that for every $a \in A$ there is a $C_a \subseteq C$ such that $C_a \vee \overline{a} \in \mathbb{C}_{t-1}$. Note that by the assumption $\mathbb{D}_{t-1} \nvDash C$ this means that if $x \in \mathit{Vars}(A) \cap \mathit{Vars}(C)$, then x occurs with the same sign in A and C, since otherwise we would get the contradiction $\mathbb{D} \vDash C \vee \overline{a} = C$. Summing up, $\mathbb{C}_{t-1}$ contains $C_a \vee \overline{a}$ for some $C_a \subseteq C$ for all $a \in \mathit{Lit}(A) \setminus \mathit{Lit}(C)$ and in addition we know that $\mathit{Lit}(A) \cap \{\overline{a} \mid a \in \mathit{Lit}(C)\} = \emptyset$. Let us write $A' = a_1 \vee \cdots \vee a_m$ and do the following weakening derivation steps from $\mathbb{C}_{t-1} \cup \{A\}$:

$$\begin{aligned} A &\rightsquigarrow C \vee A' \\ C_{a_1} \vee \overline{a}_1 &\rightsquigarrow C \vee \overline{a}_1 \\ C_{a_2} \vee \overline{a}_2 &\rightsquigarrow C \vee \overline{a}_2 \\ &\vdots \\ C_{a_m} \vee \overline{a}_m &\rightsquigarrow C \vee \overline{a}_m \end{aligned} \tag{29}$$

Then resolve $C \vee A'$ in turn with all clauses $C \vee \overline{a}_1$, $C \vee \overline{a}_2$, $\ldots$, $C_{a_m} \vee \overline{a}_m$, finally yielding the clause C.

In this way all clauses $C \in \mathbb{C}_t \setminus \mathbb{C}_{t-1}$ can be derived one by one, and we note that we never mention any variables outside of $\mathit{Vars}(\mathbb{C}_{t-1}) \cup \mathit{Vars}(A) \cup \mathit{Vars}(C)$ in these derivations.

Wrapping up the proof of Lemma 4.2, we have proven that no matter what derivation step is made in the transition $\mathbb{D}_{t-1} \rightsquigarrow \mathbb{D}_t$, we can perform the corresponding transition $\mathbb{C}_{t-1} \rightsquigarrow \mathbb{C}_t$ for our projected clause sets without the variable support size going above $\mathit{SuppSize}(\mathbb{C}_{t-1}) + \mathit{SuppSize}(\mathbb{C}_t) \leq 2 \cdot \max_{\mathbb{D} \in \pi_f}\bigl\{\mathit{SuppSize}(\mathit{proj}_F(\mathbb{D}))\bigr\}$. Also, the only time we need to download an axiom $A \in F$ in our projected refutation π of F is when π_f downloads some axiom $D \in A[f_d]$. This completes the proof of Lemma 4.2.

7 Converting Resolution Refutations of F to $\mathfrak{R}(k)$-refutations of $F[f]$

To prove part 1 of Theorem 3.3, we convert a resolution refutation π of F into a $\mathfrak{R}(d)$-refutation of the substituted formula $F[f_d]$ while (roughly) preserving the length and variable space simultaneously. This is done in two steps. First, we substitute each positive literal x appearing in a clause C in π with some d-DNF representing $f_d(\vec{x})$ and similarly substitute $\neg x$ with a d-DNF representing $\neg f_d(\vec{x})$. (Recall every function over d variables can be represented by a d-DNF formula.) The sequence of sets of clauses that was π is transformed under this substitution into a sequence of d-DNF sets that forms the "backbone" of a $\mathfrak{R}(d)$-refutation. Then, we convert the backbone into a proper $\mathfrak{R}(d)$-refutation by simulating resolution inferences and axiom downloads. Consider a resolution inference step in π which involved inferring $C \vee C'$

from $C \vee x, C \vee \neg x$. After substitution what we need to show is that $C[f_d] \vee C'[f_d]$ can be inferred from $C[f_d] \vee x[f_d], C'[f_d] \vee \neg x[f_d]$ in $\mathfrak{R}(d)$. This is shown in Lemma 7.1 below. The simulation of an axiom download is similarly addressed in Lemma 7.2, where we show that we can derive any d-DNF representation of $A[f_d]$ for an axiom $A \in F$ via a $\mathfrak{R}(k)$-derivation of bounded length and space. Given these two lemmas, the proofs of which follow below, we can complete the proof of the first part of Theorem 3.3.

Lemma 7.1 (Simulating Resolution in $\mathfrak{R}(k)$). *Suppose D_1, D_2 are two k-DNF formulas over r variables. If $D_1 \wedge D_2 \vDash 0$, then the k-DNF set $\{D_1, D_2\}$ has a $\mathfrak{R}(k)$-refutation of length $|D_1| \cdot |D_2|$ and variable space at most $2(\mathit{VarSp}(D_1) + \mathit{VarSp}(D_2))$ simultaneously.*

Lemma 7.2 (Implicational completeness of $\mathfrak{R}(k)$ with respect to clauses). *Suppose F is a CNF formula and D is a k-DNF formula and $|\mathit{Vars}(F) \cup \mathit{Vars}(D)| = r$. If $F \vDash D$ then D can be derived from F via a $\mathfrak{R}(k)$-derivation of length less than $k^{|D|} \cdot 2^{r+1}$ and variable space at most $((r+2)\,\mathit{VarSp}(D))^2$ simultaneously.*

Postponing the proofs for a moment, let us see how these two lemmas yield the first part of Theorem 3.3.

Proof of part 1 of Theorem 3.3. Let $\pi = \{\mathbb{C}_0, \dots, \mathbb{C}_\tau\}$ be a resolution refutation of the CNF formula F. Let $\pi_f = \{\mathbb{D}_0, \dots, \mathbb{D}_\tau\}$ denote the sequence of d-DNF sets obtained by substituting π with f_d in the following way. We start by fixing for each literal a a d-DNF formula representing $a[f_d]$. For a clause $C = \bigvee_i a_i$ appearing in $\mathbb{C}_t$ construct a d-DNF formula D_C which represents $C[f_d]$ by taking the disjunction of the d-DNF formulas representing a_i. Finally, set $\mathbb{D}_t = \{D_C \mid C \in \mathbb{C}_t\}$. In this way, every clause in $\mathbb{C}_t$ turns into a d-DNF formula in $\mathbb{D}_t$. Notice that the variable space of $\mathbb{D}_t$ is less than $d \cdot 2^d$ times the variable space of $\mathbb{C}_t$ because every literal appearing in $\mathbb{C}_t$ turns under substitution into a d-DNF with less than 2^d terms. To complete the proof of part 1 of Theorem 3.3 it suffices to show for $0 \leq t < \tau$ that $\mathbb{D}_{t+1}$ can be derived from $\mathbb{D}_t$ via a $\mathfrak{R}(d)$-derivation of length $\leq d^{4^{cd}} \cdot 4^{cd}$ and extra variable space $(cd+2)^3 \cdot 4^{cd} + O(1)$. We divide into cases according to the type of the t^{th} step.

Erasure If $\mathbb{C}_{t+1} = \mathbb{C}_t \setminus \{C\}$ then by construction we have $\mathbb{D}_{t+1} \subset \mathbb{D}_t$, so $\mathbb{D}_{t+1}$ can be derived in $\mathfrak{R}(d)$ from $\mathbb{D}_t$ by erasures.

Axiom download Let $A \in F$ be the axiom downloaded at time $t+1$, i.e., $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{A\}$. Let A' be an arbitrary d-DNF representation of $A[f_d]$, recalling that $A[f_d]$ is a set of axioms of $F[f_d]$. This set involves at most $c \cdot d$ many variables and $A[f_d] \vDash A'$. Furthermore, A' is a DNF formula over $2cd$ many literals so it has at most $4^{c \cdot d}$ many terms and has variable space at most $cd4^{c \cdot d}$. Applying Lemma 7.2 we conclude A' can be derived from $A[f_d]$ in length $d^{4^{cd}} \cdot 2^{cd+1}$ and variable space $(cd+2)^3 \cdot 4^{cd}$.

Inference Suppose $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{C \vee C'\}$ where $C \vee C'$ is derived from $C \vee x, C' \vee \neg x \in \mathbb{C}_t$. Notice that $(C \vee x)[f_d] = (C[f_d]) \vee x[f_d]$ and $(C' \vee \neg x)[f_d] = C'[f_d] \vee \neg x[f_d]$. Since we can bound the number of terms in a d-DNF formula representing $x[f_d]$ by 2^d, by Lemma 7.1 we can derive the empty DNF formula 0 from d-DNF formulas representing $x[f_d]$ and $\neg x[f_d]$ via a derivation of length at most 2^{2d} and variable space at most 2^{2d+1}. Applying weakening steps, when necessary, to the formulas involved in this refutation, we conclude that the d-DNF formula representing $(C \vee C')[f_d]$ can be derived from the d-DNF formulas representing $(C \vee x)[f_d]$ and $(C' \vee \neg x)[f_d]$ via a derivation of length at most 2^{2d} and 2^{2d+1} extra variable space.

Weakening Suppose $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{C \vee C'\}$ for $C \in \mathbb{C}_t$. Then the d-DNF formula representing $(C \vee C')[f_d]$ can be derived in a single step from the d-DNF formula representing $C[f_d]$ using weakening.

We have shown how to complete the conversion of π_f into a $\mathfrak{R}(d)$-refutation of $F[f_d]$ that is longer by at most a factor of $d^{4^{cd}} \cdot 2^{cd}$ and uses at most $(cd+2)^3 \cdot 4^{cd} + O(1)$ extra variable space. Taking into account the upper bound of $S \cdot d \cdot 2^d$ on the variable space of $\mathbb{D}_t$, this completes the proof of part 1 of Theorem 3.3. $\square$

It remains to prove Lemmas 7.1 and 7.2. We attend to them in order.

Proof of Lemma 7.1. First we claim that for every term $T \in D_1$ and for every term $T' \in D_2$ we have

$$T' \cap \{\neg a | a \in T\} \neq \emptyset. \tag{30}$$

To see this, assume by way of contradiction that (30) fails to hold for $T \in D_1$ and $T' \in D_2$. Consider the minimal restriction ρ that satisfies T. We see that ρ satisfies D_1 and can be extended to an assignment that satisfies T' as well, contradicting the assumption $D_1 \wedge D_2 \vDash 0$.

The refutation of $\{D_1, D_2\}$ proceeds by sequentially removing from D_1 all its terms. Let T be a term of D_1 that we wish to remove. By (30) each term $T' \in D_2$ contains a literal $\neg a$ such that $a \in T$. Apply $\wedge$-elimination to replace T' by $\neg a$. Repeating this process for each term $T' \in D_2$ we derive from D_2 in extra variable space at most $\mathit{VarSp}(D_2)$ the clause $\bigvee_{a \in T} \neg a$. Resolve this clause with D_1 to remove T. This step requires extra variable space at most $\mathit{VarSp}(D_1) + \mathit{VarSp}(D_2)$. Repeat the process for all $T \in D_1$ to obtain the empty DNF. This process required variable space at most $2(\mathit{VarSp}(D_1) + \mathit{VarSp}(D_2))$ and the refutation length is $|D_1| \cdot |D_2|$ so the lemma follows. □

Proof of Lemma 7.2. Roughly speaking, we derive from F in resolution a set of clauses that is equivalent to the k-DNF formula D. From this set of clauses we derive D using a sequence of $\wedge$-introduction inference rule applications. The key idea is to do all of this in a space-efficient manner by deriving the clauses one by one in a particular order and "merging" each derived clause into a DNF formula that, at the end of this process, turns out to be D. Details follow.

Denote $|D|$ by s. Suppose $D = \bigvee_{i=1}^{s} \bigwedge_{j=1}^{k_i} a_{i,j}$ where $k_i \leq k$ and $a_{i,j}$ denotes a literal (belonging to a set of r variables). By the distributivity of disjunction over conjunction, D is equivalent to the CNF formula

$$G_D := \bigwedge_{j_1,\ldots,j_s \in [k_1] \times \ldots \times [k_s]} \bigvee_{i=1}^{s} a_{i,j_i} \ . \tag{31}$$

Each clause of G_D is implied by F because otherwise there would be an assignment satisfying F but falsifying G_D, thereby falsifying D as well, in contradiction to the assumption $F \vDash D$. By the implicational completeness of resolution (Proposition 2.5) there is a resolution derivation of each clause of G_D from F. This derivation has length less than 2^{r+1} and space at most $(r+2)^2$ because it involves at most r variables. We now show how to construct D from the clauses of G_D.

For $s' \in [s]$ and $\vec{j} = (j_{s'+1}, \ldots, j_s) \in [k_{s'+1}] \times \ldots \times [k_s]$, let

$$D_{s',\vec{j}} = \left(\bigvee_{i=1}^{s'} \bigwedge_{j=1}^{k_i} a_{i,j} \right) \vee \bigvee_{i=s'+1}^{s} a_{i,j_i} \ . \tag{32}$$

We prove by induction on $s' \geq 0$ that $D_{s',\vec{j}}$ can be derived in variable space

$$\Big((r+2)(\mathit{VarSp}(D_{s',\vec{j}})) \Big)^2 = \left((r+2) \left(\sum_{i=1}^{s'} k_i + (s - s') \right) \right)^2 \tag{33}$$

and length less than $k^{s'} 2^{r+1}$. The base case ($s' = 0$) follows from the discussion in the previous paragraph because $D_{0,\vec{j}}$ is a single clause that is implied by F. For the inductive step assume the claim holds for $s' - 1$. We show how to derive, for $k' = 1, \ldots, k_{s'}$, the formula

$$D'_{k'} := \left(\bigvee_{i=1}^{s'-1} \bigwedge_{j=1}^{k_i} a_{i,j} \right) \vee \left(\bigwedge_{j=1}^{k'} a_{s',j} \right) \vee \bigvee_{i=s'+1}^{s} a_{i,j_i} \tag{34}$$

in length less than $k'k^{s'-1}2^{r+1}$ and variable space

$$\big((r+2)(\mathit{VarSp}(D'_{k'}))\big)^2 = \left((r+2)\left(\sum_{i=1}^{s'-1} k_i + k' + (s-s')\right)\right)^2 \quad . \tag{35}$$

This is shown by induction on $k' \geq 1$. For $k' = 1$ notice (34) is nothing but $D_{s'-1,(1,j_{s'+1},\ldots,j_s)}$ so by the inductive hypothesis with respect to $s'-1$ it can be derived in length less than $k^{s'-1}2^{r+1}$ and variable space

$$\left((r+2)\left(\sum_{i=1}^{s'-1} k_i + (s-(s'-1))\right)\right)^2 = \left((r+2)\left(\sum_{i=1}^{s'-1} k_i + k' + (s-s')\right)\right)^2 \tag{36}$$

For the inductive step assume we have derived $D'_{k'}$ using at most the variable space stated in (35). Erase all formulas in the memory but for $D'_{k'}$ and notice this remaining formula has variable space

$$\sum_{i=1}^{s'-1} k_i + k' + (s-s') \ \ . \tag{37}$$

Using the inductive hypothesis on $s'-1$ again, derive the DNF formula

$$\left(\bigvee_{i=1}^{s'-1} \bigwedge_{j=1}^{k_i} a_{i,j}\right) \vee a_{s',k'+1} \vee \bigvee_{i=s'+1}^{s} a_{i,j_i} \tag{38}$$

in variable space as in (36) and length less than $k^{s'-1}2^{r+1}$. Notice that the total variable space used is bounded by the sum given in (36) plus the sum in (37) (this latter space is required to save the formula $D'_{k'}$) so the combined variable space is at most

$$\left((r+2)\left(\sum_{i=1}^{s'-1} k_i + (s-(s'-1))\right)\right)^2 + \sum_{i=1}^{s'-1} k_i + k' + (s-s') \\ \leq \left((r+2)\left(\sum_{i=1}^{s'-1} k_i + (k'+1) + (s-s')\right)\right)^2 . \tag{39}$$

Now combine $D'_{k'}$ and (38) using a single $\bigwedge$-introduction step to obtain $D_{k'+1}$. We see that $D_{k'+1}$ can be derived in variable space bounded by (36) and length less than $k'k^{s'-1}2^{r+1}$. Summing over $k' = 1, \ldots, k$ we conclude that the derivation of $D_{s'+1,\vec{j}}$ is of length less than $k \cdot k^{s'-1}2^{r+1}$ and variable space

$$\left((r+2)\left(\sum_{i=1}^{s'+1} k_i + (s-(s'+1))\right)\right)^2 \tag{40}$$

as claimed. Setting $s' = s$ and noticing $\mathit{VarSp}(D) = \sum_{i=1}^{s} k_i$ completes the proof of the lemma. □

8 Concluding Remarks

We conclude the paper with a brief discussion of some remaining open questions.

A stronger space separation for k-DNF resolution 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 $\Omega\bigl(\sqrt[k+1]{n/\log n}\bigr)$ 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 $\Omega\bigl(\sqrt[k]{n/\log n}\bigr)$ 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 $\Omega\bigl(\sqrt[k+1]{n/\log n}\bigr)$ and $\mathrm{O}(n/\log n)$ is hence an open question.

Understanding minimally unsatisfiable k-DNF sets 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 [BSN08, BSN09] can in some sense be seen to follow from (the proof technique used to obtain) the tight bound for CNF formulas in Theorem 1.1.

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 $\mathrm{O}\bigl(\sqrt[k+1]{|\mathbb{D}|}\bigr)$ variables (Theorem 3.5) but we have no constructions of such sets with more than $\Omega(|\mathbb{D}|)$ variables (Lemma 3.6). This appears to be a natural and interesting combinatorial problem in its own right, and it would be very nice to improve the upper and/or lower bound.

Generalizations to other proof systems Our previous paper [BSN09] presented the "substitution space theorem" for resolution as a way of lifting lower bounds on the number of variables (i.e., support size) to lower bounds on (clause) space. In this paper, we extend 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 the translation in Section 4 of refutations of substitution formulas in some other proof system $\mathcal{P}$ via projection to resolution refutations of the original formula seems extremely generic and robust in that it does 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 $\mathcal{P}$). This part of the analysis is the (essentially tight) result for resolution in [BSN09, Theorem 3.12] and the (likely not tight) bound for k-DNF sets in Lemma 4.3 in this paper. 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 support size in resolution.

References

[AB04] Albert Atserias and Maria Luisa Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189(2):182–201, March 2004.

[ABE02] Albert Atserias, Maria Luisa Bonet, and Juan Luis Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Information and Computation*, 176(2):136–152, August 2002.

[ABSRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM Journal on Computing*, 31(4):1184–1211, 2002.

[Ajt88] Miklós 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, October 1988.

[AL86] Ron Aharoni and Nathan Linial. Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *Journal of Combinatorial Theory*, 43:196–204, 1986.

[Ale05] Michael 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.

[BET01] Sven Baumer, Juan Luis Esteban, and Jacobo Torán. Minimally unsatisfiable CNF formulas. *Bulletin of the European Association for Theoretical Computer Science*, 74:190–192, June 2001.

[Bla37] Archie Blake. *Canonical Expressions in Boolean Algebra*. PhD thesis, University of Chicago, 1937.

[BS02] Eli Ben-Sasson. Size space tradeoffs for resolution. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 457–464, May 2002.

[BSG03] Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003.

[BSN08] Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 709–718, October 2008.

[BSN09] Eli Ben-Sasson and Jakob Nordström. Understanding space in resolution: Optimal lower bounds and exponential trade-offs. Technical Report TR09-034, Electronic Colloquium on Computational Complexity (ECCC), March 2009.

[BSW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001.

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

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

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

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

[EGM04] Juan Luis Esteban, Nicola Galesi, and Jochen Messner. On the complexity of resolution with bounded conjunctions. *Theoretical Computer Science*, 321(2-3):347–370, August 2004.

[ET01] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001.

[HP07] Philipp Hertel and Toniann 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, October 2007.

[JN02] Jan 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, August 2002.

[Kra01] Jan Krajíček. On the weak pigeonhole principle. *Fundamenta Mathematicae*, 170(1-3):123–140, 2001.

[Kul00] Oliver 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.

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

[Nor06] Jakob Nordström. Narrow proofs may be spacious: Separating space and width in resolution (Extended abstract). In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC '06)*, pages 507–516, May 2006.

[Nor07] Jakob Nordström. A simplified way of proving trade-off results for resolution. Technical Report TR07-114, Electronic Colloquium on Computational Complexity (ECCC), September 2007.

[Nor08] Jakob Nordström. *Short Proofs May Be Spacious: Understanding Space in Resolution*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, May 2008. Available at the webpage `http://people.csail.mit.edu/jakobn/research/`.

[Nor09] Jakob Nordström. New wine into old wineskins: A survey of some pebbling classics with supplemental results. Manuscript in preparation. Current draft version available at the webpage `http://people.csail.mit.edu/jakobn/research/`, 2009.

[Raz03] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. Manuscript. Available at the webpage `http://www.mi.ras.ru/~razborov/`, July 2003.

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

[SBI04] Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM Journal on Computing*, 33(5):1171–1200, 2004.

[Seg05] Nathan Segerlind. Exponential separation between Res(k) and Res($k+1$) for $k \leq \epsilon \log n$. *Information Processing Letters*, 93(4):185–190, February 2005.