Electronic Colloquium on Computational Complexity, Report No. 34 (2009)

# Understanding Space in Resolution: Optimal Lower Bounds and Exponential Trade-offs

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

March 25, 2009

**Abstract**

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

In this paper, we remedy this situation by proving a host of length-space trade-off results for resolution in a completely general setting. Our collection of trade-offs cover space ranging over the whole interval from constant to $\mathrm{O}(n/\log\log n)$, and most of them are superpolynomial or even exponential.

Our key technical contribution is the following, somewhat surprising, theorem: Any CNF formula $F$ can be transformed by simple substitution into a new formula $F'$ such that if $F$ has the right properties, $F'$ can be proven in essentially the same length as $F$ while the minimal space needed for $F'$ is lower-bounded by the number of variables mentioned simultaneously in any proof for $F$. Applying this theorem to so-called pebbling formulas defined in terms of pebble games on directed acyclic graphs, and then using known results from the pebbling literature as well as a proving a couple of new ones, we obtain our resolution trade-off theorems.

*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.

‡Part of this work performed while at the Royal Institute of Technology (KTH) and while visiting the Technion.

ISSN 1433-8092

# 1 Introduction

## 1.1 Previous Work

**Resolution length and space** The *resolution* proof system, introduced by Blake [Bla37] in 1937 is the single-most studied proof system in propositional proof complexity. The interest in resolution is due to its lying at the very base of the important bounded-depth Frege hierarchy of propositional proof systems and because the *proof* complexity of resolution is tightly connected to the *computational* complexity of the prominent family of *SAT solvers* based on the DPLL algorithm of [DLL62, DP60, Rob65].

The interest in resolution has lead to an extensive study of the complexity of proofs in this system, which operates by refuting unsatisfiable formulas in conjunctive normal form (henceforth CNF formulas). The most important proof complexity measure is the *length* of refutations and the most important question regarding this measure has been (and still is) to establish techniques for proving lower bounds on length. Over the past half century, starting with the seminal superpolynomial lower bound for *regular* resolution by Tseitin in 1968 [Tse68], several techniques for proving superpolynomial lower bounds on this complexity measure have been discovered. Notable examples include [Hak85, Urq87, CS88, Pud97, BKPS02, BSW01, Raz03, Raz04]. We refer to the survey [Seg07] for more information on this topic.

The study of resolution *space* complexity was initiated more recently—about ten years ago—by Esteban and Torán [ET01, Tor99]. Intuitively, the space of a refutation is the maximal amount of memory needed while verifying it, and the space of refuting the CNF formula $F$ is defined as the minimal space of any resolution refutation of $F$. Over the past decade, a number of upper and lower bounds for refutation space in resolution have been presented in, for example, [ABSRW02, BSG03, EGM04, ET03].

There are two main ways to measure the amount of memory needed to verify a refutation and these measures are known as *clause space* and *variable space*. The former measure is defined as the number of different clauses in the memory, regardless of the amount of memory each clause requires. The latter is the number of literals kept in memory, i.e., it is the sum of the sizes of the clauses kept in memory. While variable space is more clearly related to the actual amount of memory required to verify a proof—the actual memory is at most $\log n$ times the variable space—clause space has attracted most of the attention. The reason for this seems to be that clause space has interesting connections to refutation length and *width*, which is the size of a largest clause in the refutation. Esteban and Torán [ET01] proved that clause space is at most logarithmic in the minimal length of a tree-like refutation of a formula, which implies that clause space is bounded by the number of variables appearing in the formula, and Atserias and Dalmau [AD03] proved that space is lower bounded by width.

The question of the relation between clause space and length of general resolution proofs was raised by the first author in [BS02] and has also been discussed in, for instance, [ET03, Seg07, Tor04]. A pair of works of the second author and Håstad [Nor06, NH08] have shown that, in contrast to the case of tree-like resolution, length and clause space of general resolution proofs are not strongly related. By this we mean that the existence of a short proof does not necessarily imply the existence of a proof that can be carried out in small clause space. In our recent joint work [BSN08] we showed that the separation of clause space and length can be "maximally" large. More precisely, the main result in our paper is an explicit construction of $k$-CNF formulas of size $n$ (for arbitrarily large $n$) that have refutations of size $O(n)$ but require clause space $\Omega(n/\log n)$. We say this separation is "maximal" because these bounds are tight up to constant factors.

**Length-space trade-offs** The focus of this paper is the fundamental question of the *trade-off* between length and space in resolution. Informally, this question asks how much time one can save when verifying a refutation by allowing more working memory during the verification process. Notice that the above-mentioned lower bounds on length and on space do not deal with this question, but rather state absolute lower bounds on each individual complexity measure. Consider for instance the maximal separation of length and space described in the previous paragraph. This separation is maximal since by combining results from [ET01, HPV77] we know that any formula refutable in time $\mathrm{O}(n)$ can also be refuted in space $\mathrm{O}(n/\log n)$. But can this linear-length refutation be carried out in space, say, $100 \cdot n/\log n$? As we show later in this paper, the answer is no in general. Sometimes short refutations require large space, and small space implies long proofs. Analogous time-space trade-offs are well-known in computational complexity (see, e.g., [CS80, CS82, LT82, Bor93, BSSV03, FLvMV05] and the survey [vM06]) and one of the main results of this work is to show how such classical time-space results can be "lifted" to give length-space trade-offs for resolution.

The question of length-space trade-offs in resolution was first studied by the first author in [BS02] and more recently by Hertel and Pitassi in [HP07] and by the second author in [Nor07]. These works have a number of limitations that are overcome in the current paper. The results of [BS02] are limited to the very restricted case of tree-like resolution. The paper [HP07] deals with variable space only and in addition require formulas with rapidly growing width, and [Nor07] uses a somewhat artificial construction of formulas "glued together" from two different unsatisfiable subformulas over disjoint variable sets. Moreover, both trade-off results for general resolution apply only for a very carefully selected ratio of space-to-formula-size and display a sharp and abrupt decay of proof length when space is increased even by small amounts. For instance, the refutation length of the formulas of [HP07] drops exponentially once the variable space is increased to $3$ literals above the bare minimal variable space required.

## 1.2 Our contribution

This paper contains two main results regarding resolution length and space, and one auxiliary result about "classical" time-space trade-offs. Our first result is a new method to obtain clause space lower bounds from lower bounds on a space measure related to variable space. The second result, which builds upon the first, is a technique to convert time-space trade-offs from the "classical" computational setting to resolution.

**The Substitution Space Theorem** To describe our first result we define the *variable support size* of a refutation as the maximal number of distinct variables appearing simultaneously in memory during the refutation. Thus, in particular, variable support size is a lower bound on variable space. We present a general method to transform lower bounds on the variable support size for $F$ to clause space lower bounds on a formula $F'$ obtained from $F$ as follows. Suppose $F$ mentions variables $x_1, \ldots, x_n$. To produce $F'$ all we do is substitute each variable $x_i$ with the exclusive-or[1] (xor) of two copies of $x_i$, denoted $x_i^{(1)}, x_i^{(2)}$ and expand the resulting "clauses" (which became disjunctions of xors after substitution) to obtain a CNF formula in the standard way. Our first main theorem can now be stated (informally) as follows.

**Theorem 1.1 (Substitution Space Theorem (Informal)).** *For any CNF formula $F$ over the set of variables $\{x_1, \ldots, x_n\}$, let $F'$ denote the formula with the exclusive-or $x_i^{(1)} \oplus x_i^{(2)}$ substituted for $x_i$, written in CNF in the canonical way.*

*Then any refutation $\pi$ of $F$ in bounded width can be transformed into a refutation $\pi'$ of $F'$ such that the length and variable space of $\pi'$ is at most a constant times the length and variable space of $\pi'$, respectively.*

*In the other direction, any refutation $\pi'$ of the substitution formula $F'$ can be translated back into a refutation $\pi$ of $F$ such that the length of $\pi$ is upper-bounded by the length of $\pi'$ and the* variable support size *of $\pi$ is at most the* clause space *of $\pi'$.*

The most surprising aspect of this theorem, which is also the hardest to prove, is that one can convert support size lower bounds for $F$ to clause space lower bounds for $F'$. This reduces tha problem of proving lower bounds on clause space to the easier task of proving lower bounds on variable support size.

[1]There is nothing magical about the exclusive-or of two variables. Substituting each variable with any function whose value is never dictated by only one variable will lead to essentially the same Substitution Space Theorem.

The proof of the Substitution Space Theorem is presented in Section 3. We believe it is of independent interest; to wit, in a subsequent work [BSN09] we generalize it to understand the connection between length and space of the stronger proof system known as $k$-DNF resolution (although the results there are weaker and apparently not tight, and in particular do not imply the results in this paper). Let us briefly describe the main ideas in the proof of the clause space–variable support size connection. A refutation $\pi'$ of $F'$ is a sequence of *clause configurations* where the $t$th configuration is a set of clauses over variables $x_1^{(1)}, x_1^{(2)}, \ldots, x_n^{(1)}, x_n^{(2)}$ corresponding to the content of the memory at time $t$ in the proof. We start by "projecting" each memory configuration down on a set of clauses over the original variables $x_1, \ldots, x_n$. Next, we argue that the sequence of projected sets is (almost) a resolution refutation of $F$, which we call $\pi$. Finally, we show that the variable support of each projected set in $\pi$ is a lower bound on the clause space of its projecting clause configuration in $\pi'$.

The Substitution Space Theorem is inspired by our recent work [BSN08] and indeed our main theorem there is a special case of this new theorem. Let us highlight the important novel aspects of this more general theorem. First and foremost, our previous statement applied only to a very special kind of formulas known as *pebbling contradictions* whereas the Substitution Space Theorem can be applied convert *any* CNF formula requiring large variable support size into a new and closely related CNF formula requiring large clause space. Second, the proof of the Substitution Space Theorem is much cleaner and simpler than the previous one. There is no longer any need to assume the existence of any "underlying directed acyclic graphs" and construct intricate intermediate resolution-like pebble games on these DAGs. Third, the Substitution Space Theorem gives length-preserving reductions from $\pi$ to $\pi'$ and vice versa, whereas it was unclear how to derive similar reductions from our previous work. And length-preserving reductions are crucial for our length-space trade-offs described below.

We end the discussion of the Substitution Space Theorem by pointing out that the space bounds obtained from the Substitution Space Theorem apply to both clause and variable space. This is because the lower bound on space of $\pi'$ is in terms of clause space. Thus, it implies a similar lower bound on the variable space of $\pi'$ because variable space is always at least as large as clause space. In the other direction, the upper bound on the space of $\pi'$ is in terms of the larger of the two space measures,variable space, and hence applies also to clause space. The "tightness of bounds" of the Substitution Space Theorem plays a pivotal role in our second main result, namely, the length-space trade-offs described next.

**Trade-offs in resolution** Our second main result is a new method to "lift" classical time-space trade-off results to the proof complexity world and obtain a host of "robust" length-space trade-offs for resolution. By "robust" we mean that the trade-off is not significantly affected by small changes to either space or time and displays a rather slow and gradual decrease in one parameter (say, length) as the other (say, space) is increased. Prior to this work such "robust" trade-offs were known only for tree-like resolution [BS02].

All trade-off results reported here follow the same proof strategy, which is described in loose terms next (the full details appear in Section 4). We start with a computational time-space trade-off which is typically stated as a result about the *pebbling price* of a directed acyclic graph. The use of pebbling in the context of space lower bounds is by now standard and we refer the reader to [Pip80] for a survey of pebbling results and to [Nor08] for a discussion of pebbling and resolution. (Relevant formal definitions appear in Section 2). The pebbling trade-off results we need are of the following nature.

> "There exists (arbitrarily large) directed acyclic graphs $G$ over $n$ vertices and bounded indegree that *(i)* can be pebbled with $p$ pebbles in time $t$, but *(ii)* any pebbling strategy of $G$ using $s < p$ pebbles requires time $f(s)$, where $f$ monotonically decreases in $s$."

One should think of $t$ as linear in $n$ and of $f(s)$ as being much larger than $t$ for small values of $s$ (We will discuss later how "large" $f(s)$ can be.)

With such a pebbling trade-off in hand, we construct from $G$ a CNF formula $F$, known as a *pebbling contradiction* (see Definition 2.11) and promptly substitute each variable by (say) the exclusive-or of two copies of the variable, as described above. Our hope is that the resulting formula, denoted $F'$, will display a length-space trade-off similar in spirit to the pebbling trade-off of the underlying graph. More to the point, the upper bound of $t$ on the time required to pebble $G$ using $p$ pebbles should imply that $F'$ can be refuted in length $\approx t$ and *variable* space $\approx p$ (consequently, the upper bound on clause space is also $\approx p$). And the Substitution Space Theorem says that a refutation $\pi'$ of $F'$ in time $t'$ and clause space $s$ implies a refutation $\pi$ of $F$ in time $\approx t'$ and variable space $\approx s$. Finally, by a close reading of the construction in [BS02], we deduce that any refutation of length $t'$ and variable support size $s$ yields a pebbling strategy for $G$ of time $t'$ and space $s$, which implies $t' > f(s)$.

Unfortunately, things are not that simple. We know how to convert a pebbling strategy into a short and space-efficient refutation only if the pebbling strategy is a so-called *black pebbling* (which corresponds to deterministic space). On the other hand, the result of [BS02] converts the proof $\pi$ into a *black-white pebbling* strategy (which corresponds to nondeterministic space). To complicate matters further, it is known that black white pebbling can be asymptotically more efficient than black pebbling [KS88, Wil85].

Thus, to obtain our trade-off results we need a strong form of "dual" pebbling trade-offs, where the upper bound *(i)* is stated in terms of *black* pebbling while the matching lower bound *(ii)* applies to the stronger model of *black-white* pebbling. Appealing to the Substitution Space Theorem, we can show that any such strong pebbling trade-off translates into a length-space trade-off for resolution.

Using this method of proof we present a number of robust size-space trade-offs for resolution. Before giving a few examples we explain why the need arises for different trade-offs (as opposed to just one global statement). In a nutshell, this is a mirror-picture of the state of size-space trade-offs for pebbling graphs upon which we rely. For instance, suppose $G$ can be pebbled in constant space. Then a straightforward counting argument shows that $G$ can be pebbled in polynomial time and constant space simultaneously. Thus, if we want to present a nontrivial size-space trade-off for a formula that can be refuted in constant space we cannot hope to get this trade-off to be superpolynomial. Similarly, if $G$ can be pebbled in, say, polylogarithmic space, we cannot obtain exponential time-space trade-offs. We are interested in deriving robust trade-offs for a large range of space complexity parameters and thus we must rely on diverse size-space trade-off results which each come from a different family of graphs. We end this section by describing a couple of trade-off results (many more appear in Section 6). We remark that all of our results are for explicitly constructible formulas.

Our strong pebbling trade-offs come from three sources. First, we prove a new strong trade-off result for a family of graphs introduced by Carlson and Savage in [CS80, CS82]. Carlson and Savage prove time-space trade-offs for these graphs in the black pebbling model, but to get a strong dual trade-off we need to modify their construction and apply different ideas to prove lower bounds in the more challenging black-white pebbling setting. (Details appear in Section 5.2.) One of the results derived from this is the rather striking statement that superpolynomial length-space trade-offs can occur for *arbitrarily slowly* growing non-constant space. (The formal statement appears as Theorem 6.2.)

**Theorem 1.2 (Superpolynomial trade-offs for super-constant space (Informal)).** *For any arbitrarily slowly growing function $s(n) = \omega(1)$ and any $\epsilon > 0$, there exists a family of $k$-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ refutable in length $\mathrm{O}(n)$ and also in space $s(n)$, but not simultaneously so. On the contrary, there are refutations of $F_n$ in simultaneous length $\mathrm{O}(n)$ and variable space $\mathrm{O}\left(\left(n/s^2(n)\right)^{1/3}\right)$, but any refutation in clause space $\mathrm{O}\left(\left(n/s^2(n)\right)^{1/3-\epsilon}\right)$ must have superpolynomial length.*

Three remarks should be made. First, notice that the trade-off applies to both clause and variable space. This is because the upper bounds are stated in terms of the larger of these two measures (variable space) while the lower bounds are in terms of the smaller one (clause space). This optimality of bound–type is inherited from the Substitution Space Theorem. Second, observe the "robust" nature of the trade-off, which is displayed by the long range of space complexity (from $\omega(1)$ up to $\approx n^{1/3}$) which requires superpolynomial length. Finally, we point out that the lower bound on length reaches up till very close to where our upper bound kicks in.

A second source of trade-off results for resolution comes from studying the graphs appearing in the study of "classical" time-space trade-offs but deriving *strictly better upper bounds on their refutation complexity than what can provably be obtained for black pebbling.* To do this, we cannot use the machinery developed in this paper as a black box, but need to prove upper bounds in resolution directly. The next theorem, a quadratic length-space trade-off for constant space, is of this type.

**Theorem 1.3 (Quadratic trade-offs for constant space (Informal)).** *There exists a family of $k$-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ refutable in length $\mathrm{O}(n)$ and also in variable space $\mathrm{O}(1)$, but not simultaneously so. On the contrary, for any refutation $\pi$ of $F_n$ in length $L$ and clause space $s$ it must hold that $L = \Omega\big((n/s)^2\big)$.*

Our third and final source of trade-off results comes from the seminal work of Lengauer and Tarjan [LT82], in which they showed strong pebbling trade-offs for variety of graphs. For instance, we can obtain the following very strong trade-off in this way.

**Theorem 1.4 (Exponential trade-offs for nearly-linear space (Informal)).** *There exists constants $K < K'$ and $\epsilon > 0$ and a family of $k$-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ that are refutable in length $\mathrm{O}(n)$ and also in variable space $K \cdot n/\log n$, but not simultaneously so. On the contrary, any refutation $\pi$ of $F_n$ in clause space $\leq K' \cdot n/\log n$ must be of length $\exp(n^{\epsilon})$.*

### 1.3 Organization of the Rest of This Paper

After a few basic definitions in Section 2, we present our first main result, the Substitution Space Theorem, in Section 3. Our second main result, namely, the method for converting strong pebbling trade-offs into length-space trade-offs for resolution, is described in Section 4. In Section 5, we derive our new pebbling trade-off and survey some previously known ones. These results are needed for the robust length-space trade-offs that are reported in in Section 6. We conclude in Section 7 with a brief discussion of some open questions.

## 2 Preliminaries

In this section we present definitions of and some basic facts about resolution and pebble games.

### 2.1 The Resolution Proof System

A *literal* is either a propositional logic variable or its negation, denoted $x$ and $\overline{x}$, respectively, or sometimes or $x^1$ and $x^0$. We define $\overline{\overline{x}} = x$. Two literals $a$ and $b$ are *strictly distinct* if $a \neq b$ and $a \neq \overline{b}$, i.e., if they refer to distinct variables.

A *clause* $C = a_1 \lor \cdots \lor a_k$ is a set of literals. Without loss of generality, all clauses $C$ are assumed to be nontrivial in the sense that all literals in $C$ are pairwise strictly distinct (otherwise $C$ is trivially true). We say that $C$ is a *subclause* of $D$ if $C \subseteq D$. A clause containing at most $k$ literals is called a $k$*-clause*.

A *CNF formula* $F = C_1 \land \cdots \land C_m$ is a set of clauses. A $k$*-CNF formula* is a CNF formula consisting of $k$-clauses. We define the *size* $S(F)$ of the formula $F$ to be the total number of literals in $F$ counted with repetitions. More often, we will be interested in the number of clauses $|F|$ of $F$.

In this paper, when nothing else is stated it is assumed that $A, B, C, D$ denote clauses, $\mathbb{C}, \mathbb{D}$ sets of clauses, $x, y$ propositional variables, $a, b, c$ literals, $\alpha, \beta$ truth value assignments and $\nu$ a truth value $0$ or $1$. We write

$$\alpha^{x=\nu}(y) = \begin{cases} \alpha(y) & \text{if } y \neq x, \\ \nu & \text{if } y = x, \end{cases} \tag{1}$$

to denote the truth value assignment that agrees with $\alpha$ everywhere except possibly at $x$, to which it assigns the value $\nu$. We let $\mathit{Vars}(C)$ denote the set of variables and $\mathit{Lit}(C)$ the set of literals in a clause $C$.[2] This notation is extended to sets of clauses by taking unions. Also, we employ the standard notation $[n] = \{1, 2, \ldots, n\}$.

In its simplest form, a *resolution derivation* $\pi : F \vdash A$ of a clause $A$ from a CNF formula $F$ can be viewed as a sequence of clauses $\pi = \{D_1, \ldots, D_\tau\}$ such that $D_\tau = A$ and each line $D_i$, $i \in [\tau]$, either is one of the clauses in $F$ (an *axiom*) or is derived from clauses $D_j, D_k$ in $\pi$ with $j, k < i$ by the *resolution rule*

$$\frac{B \vee x \quad C \vee \overline{x}}{B \vee C} \ . \tag{2}$$

We refer to (2) as *resolution on the variable* $x$ and to $B \vee C$ as the *resolvent* of $B \vee x$ and $C \vee \overline{x}$ on $x$.

When we want to study length and space simultaneously in resolution, we have to be slightly careful with the definitions so that we will be able to capture length-space trade-offs. Just listing the clauses used in a resolution refutation does not tell us *how* the refutation was performed, and essentially the same refutation can be carried out in vastly different time depending on the space constraints (as is shown in this paper). Following the exposition in [ET01], therefore, we can view a resolution refutation as a Turing machine computation, with a special read-only input tape from which the axioms can be downloaded and a working memory where all derivation steps are made. Then the length of a proof is essentially the time of the computation and space measures memory consumption. The formal definitions follow.

**Definition 2.1 (Resolution ([ABSRW02])).** A *clause configuration* $\mathbb{C}$ is a set of clauses. A sequence of clause configurations $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is a *resolution derivation* from a CNF formula $F$ if $\mathbb{C}_0 = \emptyset$ and for all $t \in [\tau]$, $\mathbb{C}_t$ is obtained from $\mathbb{C}_{t-1}$ by one of the following rules:

***Axiom Download*** $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{C\}$ for some $C \in F$ (an *axiom*).

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

***Inference*** $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{D\}$ for some $D$ inferred by resolution from $C_1, C_2 \in \mathbb{C}_{t-1}$.

A resolution derivation $\pi : F \vdash A$ of a clause $A$ from a formula $F$ is a derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ such that $\mathbb{C}_\tau = \{A\}$. A *resolution refutation*[3] of $F$ is a derivation of the empty clause $0$, i.e., the clause with no literals, from $F$. If every clause in a derivation is used at most once before being erased, we say that the derivation is *tree-like*.

For a formula $F$ and a set of formulas $\mathcal{G} = \{G_1, \ldots, G_n\}$, we say that $\mathcal{G}$ *implies* $F$, denoted $\mathcal{G} \vDash F$, if every truth value assignment satisfying all formulas $G \in \mathcal{G}$ satisfies $F$ as well. It is well known that resolution is sound and implicationally complete. That is, if there is a resolution derivation $\pi : F \vdash A$, then $F \vDash A$, and if $F \vDash A$, then there is a (tree-like) resolution derivation $\pi : F \vdash A'$ for some $A' \subseteq A$. In particular, $F$ is unsatisfiable if and only if there is a resolution refutation of $F$.

We will be interested in studying length and space in resolution, which are formalized as proof complexity measures in the next definition. Also, it will be convenient to define what width in resolution is.

[2] Although the notation $\mathit{Lit}(C)$ is slightly redundant given the definition of a clause as a set of literals, we include it for clarity.

[3] Perhaps somewhat confusingly, a resolution refutation of $F$ is sometimes also referred to as a *resolution proof* of $F$ in the literature. We will mostly stick to the term "refutation" in this paper, but will sometimes use the words "proof" and "refutation" interchangeably.

**Definition 2.2 (Length, width and space).** The *length* $L(\pi)$ of a resolution derivation $\pi$ is the total number of axiom downloads and inferences made in $\pi$, i.e., the total number of clauses counted with repetitions.

The *width* $W(C)$ of a clause $C$ is the number of literals in it, the width $W(F)$ of a formula $F$ is the size of a widest clause in $F$, and the width $W(\pi)$ of a derivation $\pi$ is defined in the same way.

The *clause space* $Sp(\mathbb{C})$ of a clause configuration $\mathbb{C}$ is $|\mathbb{C}|$, i.e., the number of clauses in $\mathbb{C}$, and the *variable space* $VarSp(\mathbb{C})$ is $\sum_{C \in \mathbb{C}} |C|$, i.e., the total number of literals in $\mathbb{C}$ counted with repetitions.[4] The clause space of a refutation $\pi$ is $Sp(\pi) = \max_{\mathbb{C} \in \pi}\{Sp(\mathbb{C})\}$ and analogously the variable space is $VarSp(\pi) = \max_{\mathbb{C} \in \pi}\{VarSp(\mathbb{C})\}$.

Taking the minimum over all refutations of a formula $F$, we define $L(F \vdash 0) = \min_{\pi : F \vdash 0}\{L(\pi)\}$ as the length of refuting $F$, $W(F \vdash 0) = \min_{\pi : F \vdash 0}\{W(\pi)\}$ as the width of refuting $F$, and $Sp(F \vdash 0) = \min_{\pi : F \vdash 0}\{Sp(\pi)\}$ and $VarSp(F \vdash 0) = \min_{\pi : F \vdash 0}\{VarSp(\pi)\}$ as the clause space and variable space, respectively, of refuting $F$ in resolution.

Note that this definition of length exactly captures the minimum length as the number of lines in a listing of the refutation (just construct a refutation that only does downloads and inferences until it gets to $0$, and only then erase all the other clauses). For tree-like resolution, we obtain the standard length measure by insisting that every clause be used at most once before being erased. Restricting the resolution derivations to tree-like resolution, we can define the measures $L_{\mathfrak{T}}(F \vdash 0)$, $Sp_{\mathfrak{T}}(F \vdash 0)$, and $VarSp_{\mathfrak{T}}(F \vdash 0)$ (note that width in general and tree-like resolution in the same, so defining tree-like width separately does not make much sense). In general, Definition 2.2 unifies previous definitions for various subsystems of resolution and gives us the possibility to measure length and space simultaneously in a meaningful way. This paper, however, will focus exclusively on general, unrestricted resolution.

Finally, we also need to define a proof complexity measure which is related to, but weaker than, variable space.[5]

**Definition 2.3 (Variable support size).** Let us say that the *variable support size*, or just *support size*, of a clause set $\mathbb{C}$ is $SuppSize(\mathbb{C}) = |Vars(\mathbb{C})|$, i.e., the number of variables mentioned in $\mathbb{C}$. We define the support size of a resolution derivation $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ to be $SuppSize(\pi) = \max_{t \in [\tau]}\{SuppSize(\mathbb{C})\}$ and the minimal support size of refuting $F$ is then $SuppSize(F \vdash 0) = \min_{\pi : F \vdash 0}\{SuppSize(\pi)\}$.

The difference between variable space and variable support size is that the variable space counts the number of variable occurrences in $\mathbb{C}$ *with repetitions*, but for variable support size we only count each variable once no matter how often it occurs. It follows that the support size of refuting a formula is always at most linear in the formula size, while the refutation variable space could potentially be quadratic in the formula size in the worst case. (It should be noted, though, that no such formulas are known to exist, and to the best of our knowledge it is even an open problem to prove superlinear lower bounds on variable space.)

## 2.2 Some Auxiliary Technical Results for Resolution

For technical reasons, it is sometimes convenient to add a derivation rule for *weakening* in resolution, saying that we can always derive a weaker clause $C' \supseteq C$ from $C$. It is easy to show that any weakening steps can always be eliminated from a refutation without changing anything essential. Let us state this more formally since we will need the precise formulation later on in this paper. The proof is a straightforward induction over the refutation and we omit the details.

[4] Note that if one wanted to be really precise, space (as well as formula size) should probably measure the number of *bits* rather than the number of literals. However, counting literals makes matters substantially cleaner, and the difference is at most a logarithmic factor. Therefore, counting literals seems to be the established way of measuring formula size and variable space.

[5] We remark that this measure has previously been studied by Hertel and Urquhart (see [Her08]), but their terminology is different in that they name this measure "variable space" and refer to variable space as "total space." While the argument can certainly be made in favour of this naming convention, we have chosen to stick with the definition of variable space used in previous papers.

**Proposition 2.4.** *Any resolution refutation* $\pi : F \vdash 0$ *using the weakening rule can be transformed into a refutation* $\pi' : F \vdash 0$ *without weakening in at most the same length, width, clause space, variable space, and support size, and performing at most the same number of axiom downloads, inferences and erasures as* $\pi$.

Another tool that we will use to to simplify some of the proofs is the concept of *restrictions*.

**Definition 2.5 (Restriction).** A *partial assignment* or *restriction* $\rho$ is a partial function $\rho : X \mapsto \{0,1\}$, where $X$ is a set of Boolean variables. We identify $\rho$ with the set of literals $\{a_1, \ldots, a_m\}$ set to true by $\rho$. The $\rho$*-restriction* of a clause $C$ is defined to be

$$C\!\restriction_\rho = \begin{cases} 1 & \text{(i.e., the trivially true clause) if } Lit(C) \cap \rho \neq \emptyset, \\ C \setminus \{\overline{a} \mid a \in \rho\} & \text{otherwise.} \end{cases}$$

This definition is extended to set of clauses by taking unions.

We write $\rho(\neg C)$ to denote the minimal restriction fixing $C$ to false, i.e., $\rho(\neg C) = \{\overline{a} \mid a \in C\}$.

**Proposition 2.6.** *If* $\pi$ *is a resolution refutation of* $F$ *and* $\rho$ *is a restriction on* $Vars(F)$*, then* $\pi\!\restriction_\rho$ *can be transformed into a resolution refutation of* $F\!\restriction_\rho$ *in at most the same length, width, clause space, variable space, and support size as* $\pi$.

In fact, $\pi\!\restriction_\rho$ *is* a refutation of $F\!\restriction_\rho$ (removing all trivially true clauses), but possibly using weakening. The proof of this is again an easy induction over the resolution refutation $\pi$.

We next state an observation that will come in handy in the proofs.

**Observation 2.7.** *Any unsatisfiable CNF formula* $F$ *over* $n$ *variables can be refuted in length at most* $2^{n+1} - 1$*, clause space at most* $\mathrm{O}(n)$*, and variable space at most* $\mathrm{O}\big(n^2\big)$ *simultaneously.*

*Proof sketch.* Build a search tree where all vertices on level $i$ query the $i$th variable and where we go to the left, say, if the variable is false under a given truth value assignment $\alpha$ and to the right if the variable is true. As soon as some axiom in $F$ 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 size at most $2^{n+1} - 1$, and if we turn it upside down we can obtain a legal tree-like refutation of $F$, possibly using weakening. This refutation can be carried out in clause space linear in the tree depth and variable space upper-bounded by the clause space times the number of distinct variables. We refer to, for instance, [BS02, ET01] for more details. ☐

In a resolution refutation of a formula $F$, there is nothing in Definition 2.1 that rules out that completely unnecessary derivation steps are made on the way, such as axioms being downloaded and them immediately erased again, or entire subderivations being made to no use. In our constructions it will be important that we can rule out some redundancies and enforce the following requirements for any resolution refutation:

- Every clause in memory is used in an inference step before being erased.
- Every clause is erased from memory immediately after having been used for the last time.

We say that a resolution refutation that meets these requirements is *frugal*. The formal definition, which is a mildly modified version of that in [BS02], follows.

**Definition 2.8 (Frugal refutation).** Let $\pi = \{\mathbb{C}_0 = \emptyset, \mathbb{C}_1, \ldots, \mathbb{C}_\tau = \{0\}\}$ be a resolution refutation of some CNF formula $F$. The *essential clauses* in $\pi$ are defined by backward induction:

- If $\mathbb{C}_t$ is the first configuration containing $0$, then $0$ is essential at time $t$.

- If $D \in \mathbb{C}_t$ is essential and is inferred at time $t$ from $C_1, C_2 \in \mathbb{C}_{t-1}$ by resolution, then $C_1$ and $C_2$ are essential at time $t-1$.

- If $D$ is essential at time $t$ and $D \in \mathbb{C}_{t-1}$, then $D$ is essential at time $t-1$.

*Essential clause configurations* are defined by forward induction over $\pi$. The configuration $\mathbb{C}_t \in \pi$ is essential if all clauses $D \in \mathbb{C}_t$ are essential at time $t$, if $\mathbb{C}_t$ is obtained by inference from a configuration $\mathbb{C}_{t-1}$ containing only essential clauses at time $t-1$, or if $\mathbb{C}_t$ is obtained from an essential configuration $\mathbb{C}_{t-1}$ by an erasure step.

Finally, $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_r\}$ is a *frugal refutation* if all configurations $\mathbb{C}_t \in \pi$ are essential.

Without loss of generality, we can always assume that resolution refutations are frugal.

**Lemma 2.9.** *Any resolution refutation $\pi : F \vdash 0$ can be converted into a frugal refutation $\pi' : F \vdash 0$ without increasing the length, width, clause space, variable space, or support size. Furthermore, the axiom downloads, inferences and erasures performed in $\pi'$ are a subset of those in $\pi$.*

*Proof.* The construction of $\pi'$ is by backward induction over $\pi$. Set $s = \min\{t : 0 \in \mathbb{C}_t\}$ and $\mathbb{C}'_s = \{0\}$. Assume that $\mathbb{C}'_s, \mathbb{C}'_{s-1}, \ldots \mathbb{C}'_{t+2}, \mathbb{C}'_{t+1}$ have been constructed and consider $\mathbb{C}_t$ and the transition $\mathbb{C}_t \rightsquigarrow \mathbb{C}_{t+1}$.

***Axiom Download*** $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{C\}$: Set $\mathbb{C}'_t = \mathbb{C}'_{t+1} \setminus \{C\}$. (If $C$ is not essential we get $\mathbb{C}'_t = \mathbb{C}'_{t+1}$.)

***Erasure*** $\mathbb{C}_{t+1} = \mathbb{C}_t \setminus \{D\}$: Ignore, i.e., set $\mathbb{C}'_t = \mathbb{C}'_{t+1}$.

***Inference*** $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{D\}$ inferred from $C_1, C_2 \in \mathbb{C}_t$: If $D \notin \mathbb{C}'_{t+1}$, ignore the step and set $\mathbb{C}'_t = \mathbb{C}'_{t+1}$. Otherwise (using fractional time steps for notational convenience) insert the configurations $\mathbb{C}'_t = \mathbb{C}'_{t+1} \cup \{C_1, C_2\} \setminus \{D\}$, $\mathbb{C}'_{t+\frac{1}{3}} = \mathbb{C}'_{t+1} \cup \{C_1, C_2\}$, $\mathbb{C}'_{t+\frac{2}{3}} = \mathbb{C}'_{t+1} \cup \{C_2\}$.

Finally go through $\pi'$ and eliminate any consecutive duplicate clause configurations.

It is straightforward to check that $\pi'$ is a legal resolution refutation. Let us verify that $\pi'$ is frugal. By backward induction, each $\mathbb{C}'_t$ for integral time steps $t$ contains only essential clauses. By forward induction, if $\mathbb{C}'_{t+1} = \mathbb{C}'_t \cup \{C\}$ is obtained by axiom download, all clauses in $\mathbb{C}'_{t+1}$ are essential. Erasures in $\pi$ are ignored. For inference steps, $\mathbb{C}'_t$ contains only essential clauses by induction, $\mathbb{C}'_{t+\frac{1}{3}}$ is essential by inference, and $\mathbb{C}'_{t+\frac{2}{3}}$ and $\mathbb{C}'_{t+1}$ are essential since they are derived by erasure from essential configurations. Finally, it is clear that $\pi'$ performs a subset of the derivation steps in $\pi$ and that the length, width, and space does not increase. □

## 2.3 Pebble Games

Pebble games were devised for studying programming languages and compiler construction, but have found a variety of applications in computational complexity theory. In connection with resolution, pebble games have been employed both to analyze resolution derivations with respect to how much memory they consume (using the original definition of space in [ET01]) and to construct CNF formulas which are hard for different variants of resolution in various respects (see for example [AJPU02, BSIW04, BEGJ00, BOP03] and the sequence of papers [Nor06, NH08, BSN08] leading up to this work). An excellent survey of pebbling up to ca. 1980 is [Pip80]. We also refer the interested reader to the upcoming survey [Nor09], which contains some later results and also describes connections between pebbling and proof complexity.

The black pebbling price of a DAG $G$ captures the memory space, i.e., the number of registers, required to perform the deterministic computation described by $G$. The space of a non-deterministic computation is measured by the black-white pebbling price of $G$. We say that vertices of $G$ with indegree 0 are *sources* and that vertices with outdegree $0$ are *sinks* (or *targets*). In the following, unless otherwise stated we will assume that all DAGs under discussion have a unique sink and this sink will always be denoted $z$. The next definition is adapted from [CS76], though we use the established pebbling terminology introduced by [HPV77].

**Definition 2.10 (Black-white pebble game).** Suppose that $G$ is a DAG with sources $S$ and a unique sink $z$. The *black-white pebble game* on $G$ is the following one-player game. At any point in the game, there are black and white pebbles placed on some vertices of $G$, at most one pebble per vertex. A *pebble configuration* is a pair of subsets $\mathbb{P} = (B, W)$ of $V(G)$, comprising the black-pebbled vertices $B$ and white-pebbled vertices $W$. The rules of the game are as follows:

1. If all immediate predecessors of an empty vertex $v$ have pebbles on them, a black pebble may be placed on $v$. In particular, a black pebble can always be placed on any vertex in $S$.
2. A black pebble may be removed from any vertex at any time.
3. A white pebble may be placed on any empty vertex at any time.
4. If all immediate predecessors of a white-pebbled vertex $v$ have pebbles on them, the white pebble on $v$ may be removed. In particular, a white pebble can always be removed from a source vertex.

A *black-white pebbling* from $(B_0, W_0)$ to $(B_\tau, W_\tau)$ in $G$ is a sequence of pebble configurations $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = (B_0, W_0)$, $\mathbb{P}_\tau = (B_\tau, W_\tau)$, and for all $t \in [\tau]$, $\mathbb{P}_t$ follows from $\mathbb{P}_{t-1}$ by one of the rules above. A *(complete) pebbling of* $G$, also called a *pebbling strategy for* $G$, is a pebbling such that $(B_0, W_0) = (\emptyset, \emptyset)$ and $(B_\tau, W_\tau) = (\{z\}, \emptyset)$.

The *time* of a pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is simply $\mathit{time}(\mathcal{P}) = \tau$ and the *space* is $\mathit{space}(\mathcal{P}) = \max_{0 \leq t \leq \tau}\{|B_t \cup W_t|\}$. The *black-white pebbling price* (also known as the *pebbling measure* or *pebbling number*) of $G$, denoted $\mathit{BW\text{-}Peb}(G)$, is the minimum space of any complete pebbling of $G$.

A *black pebbling* is a pebbling using black pebbles only, i.e., having $W_t = \emptyset$ for all $t$. The *(black) pebbling price* of $G$, denoted $\mathit{Peb}(G)$, is the minimum space of any complete black pebbling of $G$.

For any DAG $G$ over $n$ vertices with bounded indegree, the black pebbling price (and thus also the black-white pebbling price) is at most $\mathrm{O}(n/\log n)$ [HPV77], where the hidden constant depends on the indegree. A number of exact or asymptotically tight bounds on different graph families have been proven in the whole range from constant to $\Theta(n/\log n)$, for instance in [GT78, Kla85, LT80, PTC77]. As to time, obviously any DAG $G$ over $n$ vertices can be pebbled in time $2n - 1$, and for all graphs we will study this is also a lower bound, so studying the time measure in isolation is not that exciting. A very interesting question, however, is how time and space are related in a single pebbling of $G$ if one wants to optimize both measures simultaneously. We will return to this question in Section 5.

## 2.4 Pebbling Contradictions

A *pebbling contradiction* defined on a DAG $G$ is a CNF formula that encodes the pebble game on $G$ by postulating the sources to be true and the target to be false, and specifying that truth propagates through the graph according to the pebbling rules. These formulas have previously been studied in, for instance, [RM99, BEGJ00, BSW01].

**Definition 2.11 (Pebbling contradiction).** Suppose that $G$ is a DAG with sources $S$ and a unique sink $z$. Identify every vertex $v \in V(G)$ with a propositional logic variable $v$. The *pebbling contradiction* over $G$, denoted $Peb_G$, is the conjunction of the following clauses:

- for all $s \in S$, a unit clause $s$ (*source axioms*),

- For all non-source vertices $v$ with immediate predecessors $u_1, \ldots, u_\ell$, the clause $\overline{u}_1 \lor \cdots \lor \overline{u}_\ell \lor v$ (*pebbling axioms*),
- for the sink $z$, the unit clause $\overline{z}$ (*target* or *sink axiom*).

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

# 3 The Substitution Space Theorem

In this section we present the main technical contribution of this paper, the so-called Substitution Space Theorem. In order to state this theorem, we need to introduce some new definitions and notation.

## 3.1 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 $f_d(x_1, \ldots, x_d)$. For every function $f_d$, we fix some canonical representation of it as a CNF formula. We let $Cl[f_d(\vec{x})]$ denote the set of clauses in the canonical representation of $f_d$ and $Cl[\neg f_d(\vec{x})]$ denote the clauses in the canonical representation of its negation. For instance, we choose to define

$$Cl[\lor_2(\vec{x})] = \{x_1 \lor x_2\} \quad \text{and} \quad Cl[\neg\lor_2(\vec{x})] = \{\overline{x}_1, \overline{x}_2\} \tag{3}$$

for logical or of two variables and

$$Cl[\oplus_2(\vec{x})] = \{x_1 \lor x_2, \overline{x}_1 \lor \overline{x}_2\} \quad \text{and} \quad Cl[\neg\oplus_2(\vec{x})] = \{x_1 \lor \overline{x}_2, \overline{x}_1 \lor x_2\} \tag{4}$$

for logical exclusive or of two variables. The general definitions for exclusive or are

$$Cl[\oplus_d(\vec{x})] = \left\{\textstyle\bigvee_{i=1}^{d} x_i^{\nu_i} \,\middle|\, \sum_{i=1}^{d} \nu_i \equiv d \pmod 2\right\} \tag{5}$$

and

$$Cl[\neg\oplus_d(\vec{x})] = \left\{\textstyle\bigvee_{i=1}^{d} x_i^{\nu_i} \,\middle|\, \sum_{i=1}^{d} \nu_i \not\equiv d \pmod 2\right\} \tag{6}$$

from which we can see that $Cl[\oplus_d(\vec{x})]$ and $Cl[\neg\oplus_d(\vec{x})]$ both are $d$-CNFs. We will also be interested in the function saying that $k$ out of $d$ variables are true, which we will denote $k$-*true*$_d$. To give an example, for 2-*true*$_4$ we have

$$Cl[2\text{-}true_4(\vec{x})] = \left\{\begin{matrix} x_1 \lor x_2 \lor x_3, \\ x_1 \lor x_2 \lor x_4, \\ x_1 \lor x_3 \lor x_4, \\ x_2 \lor x_3 \lor x_4 \end{matrix}\right\} \tag{7}$$

and

$$Cl[\neg 2\text{-}true_4(\vec{x})] = \left\{\begin{matrix} \overline{x}_1 \lor \overline{x}_2, \\ \overline{x}_1 \lor \overline{x}_3, \\ \overline{x}_1 \lor \overline{x}_4, \\ \overline{x}_2 \lor \overline{x}_3, \\ \overline{x}_2 \lor \overline{x}_4, \\ \overline{x}_3 \lor \overline{x}_4 \end{matrix}\right\} \tag{8}$$

and in general we have

$$Cl[k\text{-}true_d(\vec{x})] = \left\{\bigvee_{i \in S} x_i \,\middle|\, S \subseteq [d],\ |S| = d - k + 1\right\} \tag{9}$$

and

$$Cl[\neg k\text{-}true_d(\vec{x})] = \left\{\bigvee_{i \in S} \overline{x}_i \,\middle|\, S \subseteq [d],\ |S| = k\right\} \ . \tag{10}$$

Clearly, $1\text{-}true_d(x_1, \ldots, x_d)$ is just another way of writing the function $\bigvee_{i=1}^{d} x_i$, and $d\text{-}true_d(x_1, \ldots, x_d) = \bigwedge_{i=1}^{d} x_i$.

In general, we could construct a canonical representation $Cl[f_d(\vec{x})]$ for $f_d$ as follows. For a truth value assignment $\alpha : \{x_1, \ldots, x_d\} \mapsto \{0, 1\}$ we define the clause $C_\alpha = x_1^{1-\alpha(x_1)} \vee \cdots \vee x_d^{1-\alpha(x_d)}$ that is true for all assignments to $x_1, \ldots, x_d$ *except* $\alpha$. Then we could define

$$Cl[f_d(\vec{x})] = \bigwedge_{\alpha \,:\, \alpha(f_d(\vec{x}))=0} C_\alpha \ . \tag{11}$$

But this way of representing the Boolean function can turn out to be unnecessarily involved. For instance, for binary logical and (11) yields $Cl[\wedge_2(\vec{x})] = \{x_1 \vee x_2,\ x_1 \vee \overline{x}_2,\ \overline{x}_1 \vee x_2\}$ instead of the arguably more natural representation $Cl[\wedge_2(\vec{x})] = \{x_1,\ x_2\}$. Therefore, we want the freedom to choose our own canonical representation when appropriate. Note, however, that (11) constitutes a proof of the fact that without loss of generality we can always assume that

$$\left| Cl[f_d(\vec{x})] \right| < 2^d \tag{12}$$

since there are only $2^d$ truth value assignments and $f_d$ is assumed to be non-constant.

The following observation is rather immediate, but nevertheless it might be helpful to state it explicitly.

**Observation 3.1.** *Suppose for any non-constant Boolean function $f_d$ that $C \in Cl[f_d(\vec{x})]$ and that $\rho$ is any partial truth value assignment such that $\rho(C) = 0$. Then for all $D \in Cl[\neg f_d(\vec{x})]$ it holds that $\rho(D) = 1$.*

*Proof.* If $\rho(C) = 0$ this means that $\rho(f_d) = 0$. Then clearly $\rho(\neg f_d) = 1$, so, in particular, $\rho$ must fix all clauses $D \in Cl[\neg f_d(\vec{x})]$ to true. ☐

We want to define formally what it means to substitute $f_d$ for the variables $Vars(F)$ in 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 $F$.

**Definition 3.2 (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 $\overline{y}$, the $f_d$-substitution is $\overline{y}[f_d] = Cl[\neg f_d(\vec{y})]$. The $f_d$-substitution of a clause $C = a_1 \vee \cdots \vee a_k$ is the CNF formula

$$C[f_d] = \bigwedge_{C_1 \in a_1[f_d]} \cdots \bigwedge_{C_k \in a_k[f_d]} (C_1 \vee \ldots \vee C_k) \tag{13}$$

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

For example, for $C = x \vee \overline{y}$ and $f_2 = x_1 \oplus x_2$ we get that

$$\begin{aligned} C[f_2] = \ & (x_1 \vee x_2 \vee y_1 \vee \overline{y}_2) \wedge (x_1 \vee x_2 \vee \overline{y}_1 \vee y_2) \\ & \wedge (\overline{x}_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2) \ . \end{aligned} \tag{14}$$

We note that $F[f_d]$ is a CNF formula over $d \cdot |Vars(F)|$ variables containing strictly less than $|F| \cdot 2^{d \cdot W(F)}$ clauses. (Recall that we defined a CNF formula as a set of clauses, which means that $|F|$ is the number of clauses in $F$.)

We have the following easy observation, the proof of which is presented for completeness.

**Observation 3.3.** *For any non-constant Boolean function $f_d : \{0,1\}^d \mapsto \{0,1\}$, it holds that $F[f_d]$ is unsatisfiable if and only if $F$ is unsatisfiable.*

*Proof.* Suppose that $F$ is satisfiable and let $\alpha$ be a truth value assignment such that $\alpha(F) = 1$. Then we can satisfy $F[f_d]$ by choosing an assignment $\alpha'$ for $Vars\big(F[f_d]\big)$ in such a way that $f_d\big(\alpha'(x_1), \ldots, \alpha'(x_d)\big) = \alpha(x)$. For if $C \in F$ is satisfied by some literal $a_i$ set to true by $\alpha$, then $\alpha'$ will satisfy all clauses $C_i \in a_i[f_d]$ and thus also the whole CNF formula $C[f_d]$ in (13).

Conversely, suppose $F$ is unsatisfiable and consider any truth value assignment $\alpha'$ for $F[f_d]$. Then $\alpha'$ defines a truth value assignment $\alpha$ for $F$ in the natural way by setting $\alpha(x) = f_d\big(\alpha'(x_1), \ldots, \alpha'(x_d)\big)$, and we know that there is some clause $C \in F$ that is not satisfied by $\alpha$. That is, for every literal $a_i \in C = a_1 \vee \cdots \vee a_k$ it holds that $\alpha(a_i) = 0$. But then $\alpha'$ does not satisfy $a_i[f_d]$, so there is some clause $C'_i \in a_i[f_d]$ such that $\alpha'(C'_i) = 0$. This shows that $\alpha'$ falsifies the disjunction $C'_1 \vee \cdots \vee C'_k \in C[f_d]$, and consequently $F[f_d]$ must also be unsatisfiable. ☐

For our present purposes, a particularly interesting kind of Boolean functions $f(x_1, \ldots, x_d)$ are those having the property that no single variable $x_i$ determines the value of $f(x_1, \ldots, x_d)$.

**Definition 3.4 (Non-authoritarian function).** We will call a Boolean function $f$ over $d$ variables $x_1, \ldots, x_d$ *non-authoritarian* if for any variable $x_i$ and any truth value $\alpha(x_i) = \nu_i$ assigned to $x_i$, $\alpha$ can be extended to a truth value assignment $\alpha'$ satisfying $f(x_1, \ldots, x_d)$ and another truth value assignment $\alpha''$ falsifying $f(x_1, \ldots, x_d)$. If $f$ does not satisfy this requirement, then we will call the function *authoritarian*.

Examples of non-authoritarian functions include exclusive-or and threshold functions over $d$ variables for which the threshold lies above $1$ and below $d$, as discussed above.

## 3.2 Formal Statement of the Theorem and Two Corollaries

Loosely put, the Substitution Space Theorem says that if a formula $F$ can be refuted in resolution in small length and width simultaneously, then so can the substitution formula $F[f_d]$. There is an analogous result in the other direction as well in the sense that we can translate any refutation $\pi_f$ of $F[f_d]$ into a refutation $\pi$ of the original formula $F$ where the length of $\pi$ is almost upper-bounded by the length of $\pi_f$ (this will be made precise below). So far this is nothing very unexpected, but what is more interesting is that if $f_d$ is non-authoritarian, then the clause space of $\pi_f$ is an upper bound on the number of variables mentioned simultaneously in $\pi$. Thus, the theorem says that we can convert lower bounds on variable support size into lower bounds on clause space by making substitutions using non-authoritarian functions.

**Theorem 3.5 (Substitution Space Theorem).** *Let $F$ be any unsatisfiable CNF formula and $f_d$ be any non-constant Boolean function of arity $d$. Then it holds that the substitution formula $F[f_d]$ can be refuted in width*

$$W\big(F[f_d] \vdash 0\big) = \mathrm{O}\big(d \cdot W(F \vdash 0)\big)$$

*and length*

$$L\big(F[f_d] \vdash 0\big) \leq \min_{\pi : F \vdash 0} \big\{ L(\pi) \cdot \exp\big(\mathrm{O}(d \cdot W(\pi))\big) \big\} \ .$$

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

Note that if $F$ is refutable simultaneously in linear length and constant width, then the bound in Theorem 3.5 on $L\big(F[f_d] \vdash 0\big)$ becomes linear in $L(F \vdash 0)$. It would be interesting to know if the bound in terms of number of axiom downloads could in fact be strengthened to a bound in terms of length, but we do not know if this is the case or not. Luckily enough, however, the bound in terms of axiom downloads turns out to be exactly what we need for our applications.

Although this might not be immediately obvious, Theorem 3.5 is remarkably powerful as a tool for understanding space in resolution. It will take some more work before we can present our main applications of this theorem, which are the strong time-space trade-off results discussed in Section 6. Let us note for starters, however, that without any extra work we immediately get lower bounds on space.

Esteban and Torán [ET01] proved that the clause space of refuting $F$ is upper-bounded by the formula size. In the papers [ABSRW02, BSG03, ET01] it was shown, using quite elaborate arguments, that there are polynomial-size $k$-CNF formulas with lower bounds on clause space matching this upper bound up to constant factors. Using Theorem 3.5 we can get a different proof of this fact.

**Corollary 3.6 ([ABSRW02, BSG03, ET01]).** *There are families of $k$-CNF formulas $\{F\}_{n=1}^{\infty}$ with $\Theta(n)$ clauses over $\Theta(n)$ variables such that $Sp(F_n \vdash 0) = \Theta(n)$.*

*Proof.* Just pick any formula family for which it is shown that any refutation of $F_n$ must at some point in the refutation mention $\Omega(n)$ variables at the same time (e.g., from [BSW01]), and then apply Theorem 3.5. ☐

It should be noted, though, that when we apply Theorem 3.5 the formulas in [ABSRW02, BSG03, ET01] are changed. We remark that there is another, and even more elegant way to derive Corollary 3.6 from [BSW01] without changing the formulas, namely by using the lower bound on clause space in terms of width in [AD03].

For our next corollary, however, there is no other, simpler way known to prove the same result. Instead, our proof in this paper actually improves the constants in the result.

**Corollary 3.7 ([BSN08]).** *There are families $\{F_n\}_{n=1}^{\infty}$ of $k$-CNF formulas of size $\mathrm{O}(n)$ refutable in linear length $L(F_n \vdash 0) = \mathrm{O}(n)$ and constant width $W(F_n \vdash 0) = \mathrm{O}(1)$ such that the minimum clause space required is $Sp(F_n \vdash 0) = \Omega(n/\log n)$.*

*Proof.* In [BS02], the first author showed that there are formulas refutable simultaneously in linear length and constant width, but for which any refutation must at some point mention $\Omega(n/\log n)$ distinct variables at the same time (although the result was stated in slightly different terms). Corollary 3.7 follows immediately from this by applying Theorem 3.5. ☐

In fact, the ideas in [BS02], which provide a way of translating back and forth between resolution and pebbling, are also what allows us to prove strong trade-off results for resolution. We will return to this in Section 4 where we formalize this resolution-pebbling correspondence.

## 3.3 Proof of the Substitution Space Theorem—Main Components

We divide the proof of Theorem 3.5 into three parts in Theorems 3.8, 3.11, and 3.12 below. In this subsection, we state these three theorems and show how they combine to yield Theorem 3.5. The rest of Section 3 is then spent proving these three auxiliary theorems.

**Theorem 3.8.** *For any CNF formula $F$ and any non-constant Boolean function $f_d$, it holds that*

$$W\big(F[f_d] \vdash 0\big) = \mathrm{O}\big(d \cdot W(F \vdash 0)\big)$$

*and*

$$L\big(F[f_d] \vdash 0\big) \leq \min_{\pi : F \vdash 0} \big\{ L(\pi) \cdot \exp\big(\mathrm{O}(d \cdot W(\pi))\big) \big\} \ .$$

These upper bounds on refutation width and length for $F[f_d]$ are not hard to show. The proof proceeds along the following lines. Given a resolution refutation $\pi$ of $F$, we construct a refutation $\pi_f : F[f_d] \vdash 0$ mimicking the derivation steps in $\pi$. When $\pi$ downloads an axiom $C$, we download the $\exp\bigl(\mathrm{O}(d \cdot W(C))\bigr)$ axiom clauses in $C[f_d]$. When $\pi$ resolves $C_1 \vee x$ and $C_2 \vee \overline{x}$ to derive $C_1 \vee C_2$, we use the fact that resolution is implicationally complete to derive $(C_1 \vee C_2)[f_d]$ from $(C_1 \vee x)[f_d]$ and $(C_2 \vee \overline{x})[f_d]$ in at most $\exp\bigl(\mathrm{O}(d \cdot W(C_1 \vee C_2))\bigr)$ steps. We return to the details of the proof in Section 3.4.

It is more challenging, however, to prove that we can get lower bounds on clause space for $F[f_d]$ from lower bounds on support size for $F$. The idea is to look at refutations of $F[f_d]$ and "project" them down on refutations of $F$. To do this, we first define a special kind of "precise implication."

**Definition 3.9 (Precise implication).** Let $F$ be a CNF formula and $f_d$ a non-constant Boolean function, and suppose that $\mathbb{D}$ is a set of clauses derived from $F[f_d]$ and that $P$ and $N$ are (disjoint) subset of variables of $F$. If

$$\mathbb{D} \vDash \bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y}) \tag{15a}$$

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

$$\mathbb{D}' \nvDash \bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y}) \ , \tag{15b}$$

$$\mathbb{D} \nvDash \bigvee_{x \in P'} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y}) \ , \text{ and} \tag{15c}$$

$$\mathbb{D} \nvDash \bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N'} \neg f_d(\vec{y}) \ , \tag{15d}$$

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

$$\mathbb{D} \rhd \bigvee_{x \in P} f_d(\vec{x}) \vee \bigvee_{y \in N} \neg f_d(\vec{y}) \ . \tag{16}$$

Note that $P = N = \emptyset$ in Definition 3.9 corresponds to $\mathbb{D}$ being unsatisfiable.

Let us also use the convention that any clause $C$ can be written $C = C^{+} \vee C^{-}$, where $C^{+} = \bigvee_{x \in Lit(C)} x$ is the disjunction of the positive literals in $C$ and $C^{-} = \bigvee_{\overline{y} \in Lit(C)} \overline{y}$ is the disjunction of the negative literals.

**Definition 3.10 (Projected clauses).** Let $F$ be a CNF formula and $f_d$ a non-constant Boolean function, and suppose that $\mathbb{D}$ is a set of clauses derived from $F[f_d]$. Then we say that $\mathbb{D}$ *projects* the clause $C = C^{+} \vee C^{-}$ on $F$—or, perhaps more correctly, on $Vars(F)$—if there is a subset $\mathbb{D}_C \subseteq \mathbb{D}$ such that

$$\mathbb{D}_C \rhd \bigvee_{x \in C^{+}} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^{-}} \neg f_d(\vec{y}) \tag{17}$$

and we write $proj_F(\mathbb{D}) = \bigl\{ C \bigm| \exists\, \mathbb{D}_C \subseteq \mathbb{D} \text{ s.t. } \mathbb{D}_C \rhd \bigvee_{x \in C^{+}} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^{-}} \neg f_d(\vec{y}) \bigr\}$ to denote the set of all clauses that $\mathbb{D}$ projects on $F$.

Given that we now know how to translate clauses derived from $F[f_d]$ into clauses over $Vars(F)$, the next step is to show that this translation preserves resolution refutations.

**Theorem 3.11.** *Suppose that $\pi_f = \bigl\{ \mathbb{D}_0, \ldots, \mathbb{D}_\tau \bigr\}$ is a resolution 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 $\bigl\{ proj_F(\mathbb{D}_0), \ldots, proj_F(\mathbb{D}_\tau) \bigr\}$ form the "backbone" of a resolution refutation $\pi$ of $F$ in the sense that:*

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

Note that by Proposition 2.4, we can get rid of the weakening moves in a postprocessing step, but allowing them in the statement of Theorem 3.11 makes the proof much cleaner. Accepting Theorem 3.11 on faith for the moment (deferring the proof to Section 3.5), the final missing link in the proof of the Substitution Space Theorem is the following lower bound.

**Theorem 3.12.** *Suppose that* $\mathbb{D} \neq \emptyset$ *is a set of clauses derived from* $F[f_d]$ *for some arbitrary unsatisfiable CNF formula* $F$ *and some non-authoritarian function* $f_d$*. Then* $\mathit{Sp}(\mathbb{D}) = |\mathbb{D}| > \mathit{SuppSize}(\mathit{proj}_F(\mathbb{D}))$.

Combining Theorems 3.8, 3.11, and 3.12, which will be proven shortly, the Substitution Space Theorem follows. This is immediate, but for the convenience of the reader we write out the details.

*Proof of Theorem 3.5.* The first part of Theorem 3.5, i.e., that any refutation $\pi$ of $F$ can be converted to a refutation $\pi_f$ of the substitution formula $F[f_d]$, is Theorem 3.8 verbatim. For the second part of Theorem 3.5, Theorem 3.11 describes how any refutation $\pi_f$ of the substitution formula $F[f_d]$ can be translated back into a refutation $\pi$ of the original formula $F$. This is true regardless of what kind of function $f_d$ is used for the substitution. If in addition $f_d$ is non-authoritarian, Theorem 3.12 says that the clause space of $\pi_f$ provides an upper bound for the variable support size of $\pi$. The theorem follows. ☐

It remains to prove Theorems 3.8, 3.11, and 3.12. For convenience of notation in the proofs, let us define the disjunction $\mathbb{C} \vee \mathbb{D}$ of two clause sets $\mathbb{C}$ and $\mathbb{D}$ to be the clause set

$$\mathbb{C} \vee \mathbb{D} = \{C \vee D \mid C \in \mathbb{C},\, D \in \mathbb{D}\} \ . \tag{18}$$

This notation extends to more than two clause sets in the natural way. Rewriting (13) in Definition 3.2 using this notation, we have that

$$(D \vee a)[f_d] = D[f_d] \vee a[f_d] = \bigwedge_{C_1 \in D[f_d]} \bigwedge_{C_2 \in a[f_d]} \big(C_1 \vee C_2\big) \ . \tag{19}$$

## 3.4 Proof of Theorem 3.8

Given $\pi : F \vdash 0$, we construct $\pi_f : D[f_d] \vdash 0$ by maintaining the invariant that if we have $\mathbb{C}$ in memory for $\pi$, then we have $\mathbb{C}[f_d]$ in memory for $\pi_f$. We get the following case analysis.

***Axiom download*** If $\pi$ downloads $C$, we download all of $C[f_d]$, i.e., less than $2^{d \cdot W(C)}$ clauses which all have width at most $d \cdot W(C)$.

***Erasure*** If $\pi$ erases $C$, we erase all of $C[f_d]$ in less than $2^{d \cdot W(C)}$ erasure steps.

***Inference*** This is the only interesting case. Suppose that $\pi$ infers $C_1 \vee C_2$ from $C_1 \vee x$ and $C_2 \vee \overline{x}$. Then by induction we have $(C_1 \vee x)[f_d]$ and $(C_2 \vee \overline{x})[f_d]$ in memory in $\pi_f$. It is a straightforward extension of Observation 3.3 that if $\mathbb{C} \vDash D$, then $\mathbb{C}[f_d] \vDash D[f_d]$, so in particular it holds that $(C_1 \vee x)[f_d]$ and $(C_2 \vee \overline{x})[f_d]$ imply $(C_1 \vee C_2)[f_d]$. By the implicational completeness of resolution, these clauses can all be derived.

An upper bound (not necessarily tight) for the width of this derivation in $\pi_f$ is $d \cdot (W(C_1 \vee x) + W(C_2 \vee \overline{x}) + W(C_1 \vee C_2)) = \mathrm{O}\big(d \cdot W(\pi)\big)$, as claimed.

To bound the length, note that $(C_1 \vee C_2)[f_d]$. contains less than $2^{d \cdot W(C_1 \vee C_2)}$ clauses. For every clause $D \in (C_1 \vee C_2)[f_d]$, consider the minimal restriction $\rho(\neg D)$ falsifying $D$. Since

$$(C_1 \vee x)[f_d] \wedge (C_2 \vee \overline{x})[f_d] \vDash D \tag{20}$$

we have that

$$(C_1 \vee x)[f_d]\!\upharpoonright_{\rho(\neg D)} \wedge (C_2 \vee \overline{x})[f_d]\!\upharpoonright_{\rho(\neg D)} \vDash 0 \ . \tag{21}$$

The number of variables is at most $d \cdot (W(C_1 \vee C_2) + 1) = N$, and by Observation 2.7 there is a refutation of $(C_1 \vee x)[f_d]\!\upharpoonright_{\rho(\neg D)} \wedge (C_2 \vee \overline{x})[f_d]\!\upharpoonright_{\rho(\neg D)}$ in length at most $2^{N+1} - 1$. Looking at this refutation and removing the restriction $\rho(\neg D)$, it is straightforward to verify that we get a derivation of $D$ from $(C_1 \vee x)[f_d] \wedge (C_2 \vee \overline{x})[f_d]$ in the same length (see, for instance, the inductive proof in [BSW01]). We can repeat this for every clause $D \in (C_1 \vee C_2)[f_d]$ to derive all of the less than $2^{d \cdot (W(C_1 \vee C_2))}$ clauses in this set in total length at most

$$2^{d \cdot (W(C_1 \vee C_2))} \cdot 2^{d \cdot (W(C_1 \vee C_2) + 2)} \leq 2^{3d \cdot W(\pi)} = 2^{\mathrm{O}(d \cdot W(\pi))} \ . \tag{22}$$

Taken together, we see that we get a refutation $\pi_f$ in length at most $L(\pi) \cdot 2^{\mathrm{O}(d \cdot W(\pi))}$ and width at most $\mathrm{O}\big(d \cdot W(\pi)\big)$. Theorem 3.8 follows.

## 3.5 Proof of Theorem 3.11

Let us use the convention that $\mathbb{D}$ and $D$ denote clause sets and clauses 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 $\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})$ (with precise implication $\rhd$ defined as in Definition 3.9). Note that it will always be clear when we use the notation in this overloaded sense since $\mathbb{D}$ and $C$ are defined over different sets of variables, and so the non-overloaded interpretation would not be very meaningful.

Recall from Definition 3.10 that $\mathit{proj}_F(\mathbb{D}) = \big\{ C \,\big|\, \exists\, \mathbb{D}_C \subseteq \mathbb{D} \text{ s.t. } \mathbb{D}_C \rhd \bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y}) \big\}$ is the set of clauses projected by $\mathbb{D}$. In the spirit of the notational convention just introduced, we will let $\mathbb{C}_t$ be a shorthand for $\mathit{proj}_F(\mathbb{D}_t)$.

Suppose now that $\pi_f = \big\{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\big\}$ is a 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 Theorem 3.11 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 tricky 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 $\max\big\{\mathit{SuppSize}(\mathbb{C}_{t-1}),\ \mathit{SuppSize}(\mathbb{C}_t)\big\}$ during the intermediate derivation steps needed in $\pi$. 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 3.13.** *Using the overloaded notation just introduced, 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 $\mathbb{D}' \subseteq \mathbb{D}_t$, $C_1^+ \subseteq C^+$, 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 $\pi_f$ 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.

### 3.5.1 Inference

Since $\mathbb{D}_t \supseteq \mathbb{D}_{t-1}$, it is immediate from Definition 3.10 that no clauses in $\mathbb{C}_{t-1}$ can disappear at time $t$, i.e., $\mathbb{C}_{t-1} \setminus \mathbb{C}_t = \emptyset$. There can appear new clauses in $\mathbb{C}_t$, but by Observation 3.13 all such clauses are derivable by weakening from $\mathbb{C}_{t-1}$. During such weakening moves the variable support size increases monotonically and is bounded from above by $\mathit{SuppSize}(\mathbb{C}_t)$.

### 3.5.2 Erasure

Since $\mathbb{D}_{t-1} \subseteq \mathbb{D}_t$, it is immediate from Definition 3.10 that no new clauses can appear at time $t$. Any clauses in $\mathbb{C}_{t-1} \setminus \mathbb{C}_t$ can simply be erased, which decreases the variable support size monotonically.

### 3.5.3 Axiom download

This is the only place in the case analysis where we need to do some 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, $D$ must be involved in projecting it so there is some subset $\mathbb{D} \subseteq \mathbb{D}_{t-1}$ such that

$$\mathbb{D} \cup \{D\} \rhd C \ . \tag{23}$$

Also note that if $\mathbb{D}_{t-1} \vDash C$ we are done since $C$ can be derived from $\mathbb{C}_{t-1}$ by weakening, so we can assume that

$$\mathbb{D}_{t-1} \nvDash C \ . \tag{24}$$

We want to show that all clauses $C$ satisfying (23) and (24) 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 $\max\{\mathit{SuppSize}(\mathbb{C}_{t-1}), \mathit{SuppSize}(\mathbb{C}_t)\}$. The key to our proof is the next lemma.

**Lemma 3.14.** *Suppose that* $\mathbb{D}$ *derived from* $F[f_d]$, $D \in A[f_d]$, *and* $C$ *a clause over* $\mathit{Vars}(F)$ *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 clause subset* $\mathbb{D}^i \subseteq \mathbb{D}$ *and a subclause* $C^i \subseteq C$ *such that* $\mathbb{D}^i \rhd C^i \vee \overline{a}_i$. *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 truth value assignment $\alpha$ such that $\alpha(\mathbb{D}) = 1$ but $\alpha(\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})) = 0$. Such an assignment exists since $\mathbb{D} \nvDash C$ by assumption. Also, since by assumption $\mathbb{D} \cup \{D\} \rhd C$ we must have $\alpha(D) = 0$. If $A = a_1 \vee \cdots \vee a_k$, we can write $D \in A[f_d]$ on the form $D = D_1 \vee \cdots \vee D_k$ for $D_i \in a_i[f_d]$ (compare with (19)). Fix any $a \in A$ and suppose for the moment that $a = x$ is a positive literal. Then $\alpha(D_i) = 0$ implies that $\alpha(f_d(\vec{x})) = 0$. By Observation 3.1, this means that $\alpha(\neg f_d(\vec{x})) = 1$. Since exactly the same argument holds if $a = \overline{y}$ is a negative literal, we conclude that

$$\mathbb{D} \vDash \bigvee_{x \in (C \vee \overline{a}_i)^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in (C \vee \overline{a}_i)^-} \neg f_d(\vec{y}) \tag{25}$$

or, rewriting (25) using our overloaded notation, that

$$\mathbb{D} \vDash C \vee \overline{a}_i \quad . \tag{26}$$

If $a_i \in C$, the clause $C \vee \overline{a}_i$ is trivially true and thus uninteresting, but otherwise we pick $\mathbb{D}^i \subseteq \mathbb{D}$ and $C^i \subseteq C$ minimal such that (26) still holds (and notice that since $\mathbb{D} \nvDash C$, the literal $\overline{a}_i$ cannot be dropped from the implication). Then by Definition 3.10 we have $\mathbb{D}^i \rhd C^i \vee \overline{a}_i$ as claimed. ☐

We remark that Lemma 3.14 can be seen to imply that $\mathit{Vars}(A) \subseteq \mathit{Vars}(\mathbb{C}_t) = \mathit{Vars}(\mathit{proj}_F(\mathbb{D}_t))$. For $x \in \mathit{Vars}(A) \cap \mathit{Vars}(C)$ this is of course trivially true, but for $x \in \mathit{Vars}(A) \setminus \mathit{Vars}(C)$ Lemma 3.14 tells us that already at time $t-1$, there is a clause in $\mathbb{C}_{t-1} = \mathit{proj}_F(\mathbb{D}_{t-1})$ containing $x$, namely the clause $C^i \vee \overline{a}_i$ found in the proof above. Since $\mathbb{D}_t \supseteq \mathbb{D}_{t-1}$, this clause does not disappear at time $t$. This means that if we download $A \in F$ in our refutation $\pi : F \vdash 0$ under construction, we have $\mathit{SuppSize}(\mathbb{C}_{t-1} \cup \{A\}) \leq \mathit{SuppSize}(\mathbb{C}_t)$. 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 $\max\{\mathit{SuppSize}(\mathbb{C}_{t-1}), \mathit{SuppSize}(\mathbb{C}_t)\}$.

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 3.14 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 assumption (24) 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 $\mathbb{C}_{t-1} \cup \{A\}$:

$$\begin{aligned} A &\rightsquigarrow C \vee A' \\ C_{a_1} \vee \overline{a}_1 &\rightsquigarrow C \vee \overline{a}_1 \\ C_{a_2} \vee \overline{a}_2 &\rightsquigarrow C \vee \overline{a}_2 \\ &\vdots \\ C_{a_m} \vee \overline{a}_m &\rightsquigarrow C \vee \overline{a}_m \end{aligned} \tag{27}$$

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 $\mathit{Vars}(\mathbb{C}_{t-1} \cup \{A\}) \subseteq \mathit{Vars}(\mathbb{C}_t)$ in these derivations.

### 3.5.4 Wrapping up the Proof of Theorem 3.11

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 $\max\{\mathit{SuppSize}(\mathbb{C}_{t-1}), \mathit{SuppSize}(\mathbb{C}_t)\}$. Also, the only time we need to download an axiom $A \in F$ in our projected refutation $\pi$ of $F$ is when $\pi_f$ downloads some axiom $D \in A[f_d]$. This completes the proof of Theorem 3.11.

## 3.6 Proof of Theorem 3.12

Recall the convention that $x, y, z$ refer to variables in $F$ while $x_1, \ldots, x_d, y_1, \ldots, y_d, z_1, \ldots, z_d$ refer to variables in $F[f_d]$. Also recall that we use overloaded notation $\mathbb{D} \vDash C$, $\mathbb{D} \nvDash C$, and $\mathbb{D} \rhd C$ for $C = C^+ \vee C^-$ (where $C^+ = \bigvee_{x \in C} x$ and $C^- = \bigvee_{\overline{y} \in C} \overline{y}$) when the corresponding implications hold or do not hold for $\mathbb{D}$ with respect to $\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})$.

We start with an intuitively plausible lemma saying that for all variables $x$ appearing in some clause projected by $\mathbb{D}$, the clause set $\mathbb{D}$ itself must contain at least one of the variables $x_1, \ldots, x_d$.

**Lemma 3.15.** *Suppose that $\mathbb{D}$ is a set of clauses derived from $F[f_d]$ and that $C \in \mathit{proj}_F(\mathbb{D})$. Then for all variables $x \in \mathit{Vars}(C)$ it holds that $\{x_1, \ldots, x_d\} \cap \mathit{Vars}(\mathbb{D}) \neq \emptyset$.*

*Proof.* Fix any $\mathbb{D}' \subseteq \mathbb{D}$ such that $\mathbb{D}$ implies $C$ precisely in the sense of Definition 3.9. By this definition, for all $z \in \mathit{Vars}(C)$ we have $\mathbb{D}' \nvDash C \setminus \{z, \overline{z}\}$. Suppose that $z$ appears as a positive literal in $C$ (the case of a negative literal is completely analogous). This means that there is an assignment $\alpha$ such that $\alpha(\mathbb{D}') = 1$ but $\alpha\bigl(\bigvee_{x \in C^+ \setminus \{z\}} f_d(\vec{x}) \vee \bigvee_{y \in C^-} \neg f_d(\vec{y})\bigr) = 0$. Since $\mathbb{D}' \rhd C$, it must hold that $\alpha(f_d(\vec{z})) = 1$. Modify $\alpha$ into $\alpha'$ by changing the assignments to $z_1, \ldots, z_d$ in such a way that $\alpha'(f_d(\vec{z})) = 0$. Then $\alpha'\bigl(\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{y \in C^-} \neg f_d(\vec{y})\bigr) = 0$, so we must have $\alpha'(\mathbb{D}') = 0$. Since we only changed the assignments to (a subset of) the variables $z_1, \ldots, z_d$, the clause set $\mathbb{D}' \subseteq \mathbb{D}$ must mention at least one of these variables. ☐

With Lemma 3.15 in hand, we are ready to prove Theorem 3.12. Note that everything said so far in Section 3 (in particular, all of the proofs) applies to any non-constant Boolean function. In the proof of Theorem 3.12, however, it will be essential that we are dealing with non-authoritarian functions, i.e., functions $f_d$ having the property that no single variable $x_i$ can fix the the value of $f_d(x_1, \ldots, x_d)$.

Suppose that $\mathbb{D}$ is a set of clauses derived from $F[f_d]$ and write $V^* = \mathit{Vars}(\mathit{proj}_F(\mathbb{D}))$ to denote the set of all variables in $\mathit{Vars}(F)$ appearing in any clause projected by $\mathbb{D}$. We want to prove that $\mathit{Sp}(\mathbb{D}) = |\mathbb{D}| > \bigl|V^*\bigr|$ provided that $f_d$ is non-authoritarian.

To this end, consider the bipartite graph with the clauses in $\mathbb{D}$ labelling the vertices on the left-hand side and variables in $V^*$ labelling the vertices on the right-hand side. We draw an edge between $D \in \mathbb{D}$ and $x \in V^*$ if $\mathit{Vars}(D) \cap \{x_1, \ldots, x_d\} \neq \emptyset$. By Lemma 3.15 it holds that $\mathit{Vars}(\mathbb{D}) \cap \{x_1, \ldots, x_d\} \neq \emptyset$ for all variables $x \in V^*$, so in particular every variable $x \in V^*$ is the neighbour of at least one clause $D \in \mathbb{D}$. Let us write $N(D)$ to denote the neighbours of a left-hand vertex $D$ and extend this notation to sets of vertices by taking unions.

We claim that if $V^* = \mathit{Vars}(\mathit{proj}_F(\mathbb{D})) \neq \emptyset$, then there must exist some clause set $\mathbb{D}' \subseteq \mathbb{D}$ satisfying $|\mathbb{D}'| > N(\mathbb{D}')$. Suppose on the contrary that $|\mathbb{D}'| \leq N(\mathbb{D}')$ for all $\mathbb{D}' \subseteq \mathbb{D}$. Then by Hall's marriage theorem there is a matching of the clauses in $\mathbb{D}$ into the variable set $V^*$. Assume that $C = C^+ \vee C^-$ is any clause projected by $\mathbb{D}$ (such a clause exists since $V^* \neq \emptyset$). Then surely

$$\mathbb{D} \vDash \bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y}) \tag{28}$$

(there is even a subset of $\mathbb{D}$ such that this implication is precise). But using the matching between $\mathbb{D}$ and $V^*$, we can satisfy $\mathbb{D}$ without assigning values to more than one variable $x_i \in \mathit{Vars}(\mathbb{D})$ corresponding to any $x \in \mathit{Vars}(F)$. Since $f_d$ is non-authoritarian, we can then extend this assignment to another assignment falsifying $f_d(\vec{x})$ for all $x \in C^+$ and satisfying $f_d(\vec{y})$ for all $\overline{y} \in C^-$. This means that our assignment satisfies the left-hand side of the implication (28) but falsifies the right-hand side, which is a contradiction. The claim follows.

Hence, fix any largest subset $\mathbb{D}_1 \subseteq \mathbb{D}$ such that $|\mathbb{D}_1| > N(\mathbb{D}_1)$. Clearly, if $\mathbb{D}_1 = \mathbb{D}$ we are done (remember that $N(\mathbb{D}) = V^*$), so suppose $\mathbb{D}_1 \neq \mathbb{D}$. In much the same way as above, we show that this assumption leads to a contradiction.

Let $\mathbb{D}_2 = \mathbb{D} \setminus \mathbb{D}_1 \neq \emptyset$ and define the vertex sets $V_1^* = N(\mathbb{D}_1)$ and $V_2^* = V^* \setminus V_1^*$. Note that we must have $V_2^* \subseteq N(\mathbb{D}_2)$ since $N(\mathbb{D}) = N(\mathbb{D}_1) \cup N(\mathbb{D}_2) = V^*$. By the maximality of $\mathbb{D}_1$ it must hold for all $\mathbb{D}' \subseteq \mathbb{D}_2$ that $|\mathbb{D}'| \leq \bigl|N(\mathbb{D}') \setminus V_1^*\bigr|$, because otherwise $\mathbb{D}'' = \mathbb{D}_1 \cup \mathbb{D}'$ would be a larger set with $|\mathbb{D}''| > |N(\mathbb{D}'')|$. But this implies that, again by Hall's marriage theorem, there is a matching $M$ of $\mathbb{D}_2$ into

$N(\mathbb{D}_2) \setminus V_1^* = V_2^*$. Consider any clause $C \in \mathit{proj}_F(\mathbb{D})$ such that $\mathit{Vars}(C) \cap V_2^* \neq \emptyset$ and let $\mathbb{D}' \subseteq \mathbb{D}$ be any clause set such that

$$\mathbb{D}' \rhd \bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y}) \tag{29}$$

(the existence of which is guaranteed by Definition 3.10). We claim that we can construct an assignment $\alpha$ that makes $\mathbb{D}'$ true but $\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})$ false. This is clearly a contradiction, so if we can prove this claim it follows that our assumption $\mathbb{D}_1 \neq \mathbb{D}$ is false and that it instead must hold that $\mathbb{D}_1 = \mathbb{D}$ and thus $\left|N(\mathbb{D})\right| = \left|V^*\right| < |\mathbb{D}|$, which proves the theorem.

To establish the claim, let $\mathbb{D}'_i = \mathbb{D}' \cap \mathbb{D}_i$ for $i = 1, 2$ and let $C_i = C_i^+ \vee C_i^-$ for

$$C_i^+ = \bigvee_{\substack{x \in C \\ x \in V_i^*}} x \quad \text{and} \quad C_i^- = \bigvee_{\substack{\overline{y} \in C \\ y \in V_i^*}} \overline{y} \tag{30}$$

and $i = 1, 2$. We construct the assignment $\alpha$ satisfying $\mathbb{D}'$ but falsifying $\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})$ in three steps:

1. Since $C_1^+ \vee C_i^- = C_1 \subsetneq C$ by construction (recall that we chose our clause $C$ in such a way that $\mathit{Vars}(C) \cap V_2^* \neq \emptyset$), the minimality condition in Definition 3.10 yields that

$$\mathbb{D}'_1 \nvDash \bigvee_{x \in C_1^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C_1^-} \neg f_d(\vec{y}) \tag{31}$$

   and hence we can find a truth value assignment $\alpha_1$ that sets $\mathbb{D}'_1$ to true, all $f_d(x_1, \ldots, x_d)$, $x \in C_1^+$, to false, and all $f_d(y_1, \ldots, y_d)$, $\overline{y} \in C_1^-$, to true. Note that $\alpha_1$ need only assign values to $\{z_1, \ldots, z_d \mid z \in \mathit{Vars}(C_1)\}$.

2. For $\mathbb{D}'_2$, we use the matching $M$ into $V_2^*$ found above to pick a distinct variable $x(D) \in \mathit{Vars}(F)$ for every $D \in \mathbb{D}'_2$ and then a variable $x(D)_i \in \mathit{Vars}(F[f_d])$ appearing in $D$, the existence of which is guaranteed by the edge between $D$ and $x(D)$. Let $\alpha_2$ be the assignment that sets all these variables $x(D)_i$ to the values that fix all $D \in \mathbb{D}'_2$ to true. We stress that $\alpha_2$ assigns a value to at most one variable $x(D)_i$ for every $x(D) \in \mathit{Vars}(F)$.

3. But since $f_d$ is non-authoritarian, this means that we can extend $\alpha_2$ to an assignment to all variables $x(D)_1, \ldots, x(D)_d$ that still satisfies $\mathbb{D}'_2$ but sets all $f_d(x_1, \ldots, x_d)$, $x \in C_2^+$, to false and all $f_d(y_1, \ldots, y_d)$, $\overline{y} \in C_2^-$, to true.

Hence, $\alpha = \alpha_1 \cup \alpha_2$ is an assignment such that $\alpha(\mathbb{D}') = 1$ but $\alpha(\bigvee_{x \in C^+} f_d(\vec{x}) \vee \bigvee_{\overline{y} \in C^-} \neg f_d(\vec{y})) = 0$, which proves the claim. This concludes the proof of Theorem 3.12.

Since Theorems 3.8, 3.11, and 3.12 have now all been established, the proof of Theorem 3.5 is finished.

# 4 Reductions Between Resolution and Pebbling

It is not hard to see how a black pebbling $\mathcal{P}$ of a DAG $G$ can be used to construct a resolution refutation of the pebbling contradiction $\mathit{Peb}_G$ in Definition 2.11 in length and space upper-bounded by $\mathsf{time}(\mathcal{P})$ and $\mathsf{space}(\mathcal{P})$, respectively. It is straightforward to show that this translation from pebblings to refutations works even if we do an $f_d$-substitution in the pebbling contradiction. We present a proof of this fact in Section 4.1.

Using our new results in Section 3, we can prove the more surprising fact that there is also a fairly tight reduction in the other direction: provided that the function $f_d$ is non-authoritarian, any resolution refutation of $\mathit{Peb}_G[f_d]$ translates into a black-white pebbling of $G$ with the same time-space properties (adjusting for constant factors depending on the function $f_d$ and the maximal indegree of $G$). This new reduction is given in Section 4.2.

Finally, in Section 4.3 we appeal to both of these reductions to prove a meta-theorem saying that for DAGs $G$ having the right time-space trade-off properties, we can prove that pebbling contradictions defined over such DAGs inherit the same trade-off properties. This will allow us, after having studied pebbling time-space trade-offs in Section 5, to prove a wealth of strong trade-offs for both clause space and variable space in resolution in Section 6.

## 4.1 From Black Pebblings to Resolution Refutations

Given any black-only pebbling $\mathcal{P}$ of a DAG $G$, we can mimic this pebbling in a resolution refutation of $Peb_G$ by deriving that a literal $v$ is true whenever the corresponding vertex in $G$ is pebbled (this was perhaps first observed in [BSIW04]). This construction carries over also to substitution formulas $Peb_G[f_d]$ and we have the following theorem.

**Theorem 4.1.** *Let $f_d$ be a non-constant Boolean function of arity $d$ and let $G$ be a DAG with indegree at most $\ell$ and unique sink $z$. Then given any complete black pebbling $\mathcal{P}$ of $G$, we can construct a resolution refutation $\pi : Peb_G[f_d] \vdash 0$ such that*

$$\begin{aligned} L(\pi) &\leq \textit{time}(\mathcal{P}) \cdot \exp\bigl(\mathrm{O}(d(\ell+1))\bigr) \ , \\ W(\pi) &\leq d(\ell+1) \ , \textit{ and} \\ \mathit{VarSp}(\pi) &\leq \textit{space}(\mathcal{P}) \cdot \exp\bigl(\mathrm{O}(d(\ell+1))\bigr) \ . \end{aligned}$$

Before presenting the proof, we note that in our applications we will have the function arity $d$ and the DAG indegree $\ell$ fixed (we can for instance pick $d = \ell = 2$), which means that the bounds on length and space above turns into $L(\pi) = \mathrm{O}\bigl(\textit{time}(\mathcal{P})\bigr)$ and $\mathit{VarSp}(\pi) = \mathrm{O}\bigl(\textit{space}(\mathcal{P})\bigr)$. We also remark that for concrete functions $f_d$, such as for instance XOR over two variables, we can easily compute explicit upper bounds on the constants hidden in the asymptotic notation if we so wish, and these constants are small.

*Proof of Theorem 4.1.* The proof is by induction over the black pebbling $\mathcal{P}$. We maintain the invariant that if at time $t$ we have black pebbles on the vertices in $V$, then $\pi$ will contain exactly the clauses $\mathbb{C}_t = \{x[f_d] \mid x \in V\}$. To simplify the notation in the proof, we will again use fractional time steps in $\pi$, making sure that it never takes more than $\exp\bigl(\mathrm{O}(d(\ell+1))\bigr)$ time steps to get from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$.

Consider the pebbling move made in $\mathcal{P}$ at time $t$ :

1. If $\mathcal{P}$ places a pebble on a source vertex $s$, we download the less than $2^d$ axioms in $s[f_d]$.

2. If $\mathcal{P}$ places a pebble on a non-source vertex $v$ with immediate predecessors $u_1, \ldots, u_{\ell'}$, by induction we have $\{u_i[f_d] \mid i = 1, \ldots, \ell'\} \subseteq \mathbb{C}_{t-1}$. The argument in this case is very similar to the one in Section 3.4.

   First download the less than $2^{d(\ell'+1)}$ pebbling axioms in $(\overline{u}_1 \lor \cdots \lor \overline{u}_{\ell'} \lor v)[f_d]$. Now

   $$\{u_i[f_d] \mid i = 1, \ldots, \ell'\} \cup \{(\overline{u}_1 \lor \cdots \lor \overline{u}_{\ell'} \lor v)[f_d]\} \tag{32}$$

   implies all clauses $D \in v[f_d]$. If we apply the restriction $\rho(\neg D)$ to the clause set (32) we can obtain a refutation in length and variable space at most $\exp\bigl(\mathrm{O}(d(\ell+1))\bigr)$ (and trivially in width at most $d(\ell+1)$) by Observation 2.7. Removing the restriction $\rho(\neg D)$, this refutation turns into a derivation of $D$. Doing this for all of the less than $2^d$ clauses $D \in v[f_d]$ completes the induction step.

3. If $\mathcal{P}$ removes a pebble from any vertex $v$, we erase the less than $2^d$ clauses in $v[f_d]$ from memory.

At the end of the pebbling $\mathcal{P}$, we have $\mathbb{C}_\tau = \{z[f_d]\}$ for $z$ the sink of $G$. We conclude the refutation by downloading all the sink axioms in $\overline{z}[f_d]$ and deriving the empty clause $0$ in length $\exp(\mathrm{O}(d))$, width $d$ and variable space $\exp(\mathrm{O}(d))$. ☐

## 4.2 From Resolution Refutations to Black-White Pebblings

Let us now see how we can go in the other direction from resolution refutations to pebbling strategies.

**Theorem 4.2.** *Let $f$ be any non-authoritarian Boolean function and $G$ be any DAG with unique sink and bounded indegree $\ell$. Then from any resolution refutation $\pi : Peb_G[f] \vdash 0$ we can extract a black-white pebbling strategy $\mathcal{P}_\pi$ for $G$ such that $\mathsf{time}(\mathcal{P}_\pi) \leq (\ell + 1) \cdot L(\pi)$ and $\mathsf{space}(\mathcal{P}_\pi) \leq Sp(\pi)$.*

Before proving this theorem, we want to stress that Theorems 4.1 and 4.2 are not perfect converses. This is so since the reduction in one direction uses black pebbling (Theorem 4.1) while the reduction in the other direction is in terms of black-white pebbling (Theorem 4.2) and there can be a quadratic difference in pebbling price depending on whether white pebbles may be used or not [KS88]. The problem here is that we do not know of any way of translating black-white pebbling strategies into resolution refutations that preserve the time and space properties. Indeed, we believe that this is not a mere technicality but that it is in fact not possible in general to convert black-white pebblings to resolution refutations with the same time-space trade-off properties. Formalizing and proving such a statement is another matter, however, and we leave it as an open problem.

The proof of Theorem 4.2 is in three steps:

1. First, we convert $\pi : Peb_G[f] \vdash 0$ to a refutation $\pi'$ of $Peb_G$ such that $SuppSize(\pi') \leq Sp(\pi)$ and the number of axiom downloads in $\pi'$ is upper-bounded by the number of axiom downloads in $\pi$. This is Theorem 3.5, which is the key technical contribution of this paper.

2. The refutation $\pi' : Peb_G \vdash 0$ can contain weakening moves, which we do not want, so we appeal to Proposition 2.4 to get a refutation $\pi'' : Peb_G \vdash 0$ without any weakening steps. By Lemma 2.9, without loss of generality we can assume that $\pi''$ is frugal (Definition 2.8). This part of the proof just uses standard techniques, and the number of axiom downloads and the variable support size can only decrease when going from $\pi'$ to $\pi''$.

3. Finally, we show that $\pi''$ corresponds to a black-white pebbling strategy $\mathcal{P}$ for $G$ such that $\mathsf{time}(\mathcal{P})$ is upper-bounded by the number of axiom downloads and $\mathsf{space}(\mathcal{P})$ by the maximal number of variables occurring simultaneously in $\pi''$. This final part relies heavily on the work [BS02] by the first author. Since we need a more detailed result than can be read off from that paper, however, we present the full construction below.

Putting together these three steps, Theorem 4.2 clearly follows. What remains is thus to prove the following lemma.

**Lemma 4.3.** *Let $G$ be any DAG with unique sink and bounded indegree $\ell$, and suppose that $\pi$ is any resolution refutation of $Peb_G$ without weakening that is also frugal. Then there is a black-white pebbling strategy $\mathcal{P}_\pi$ for $G$ such that $\mathsf{space}(\mathcal{P}_\pi) \leq SuppSize(\pi)$ and $\mathsf{time}(\mathcal{P}_\pi)$ is at most $(\ell + 1)$ times the number of axiom downloads in $\pi$.*

*Proof.* Given a refutation $\pi = \{\mathbb{C}_0 = \emptyset, \mathbb{C}_1, \ldots, \mathbb{C}_\tau = \{0\}\}$ of $Peb_G$, we translate every clause set $\mathbb{C}_t$ into a black-white pebble configuration $\mathbb{P}_t = (B_t, W_t)$ using a slightly modified version of the ideas in [BS02], and then show that $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is essentially a legal black-white pebbling of $G$ as in the statement of the lemma. The translation will satisfy the invariant that $B_t \cup W_t = Vars(\mathbb{C}_t)$ which yields the upper bound on space in terms of variable support size. The first configuration $\mathbb{C}_0 = 0$ is thus translated into $\mathbb{P}_0 = (\emptyset, \emptyset)$.

Suppose inductively that $(B_{t-1}, W_{t-1})$ has been constructed from $\mathbb{C}_{t-1}$ and consider all the variables $x \in \mathit{Vars}(\mathbb{C}_t)$ one by one. If $x \in \mathit{Vars}(\mathbb{C}_t) \cap B_{t-1}$, keep $x$ in $B_t$. Otherwise, if $x \in \mathit{Lit}(\mathbb{C}_t)$ appears as a positive literal, add $x$ to $B_t$. Otherwise, if $\overline{x} \in \mathit{Lit}(\mathbb{C}_t)$, add $x$ to $W_t$. This is our translation of $\mathbb{C}_t$ into black pebbles $B_t$ and white pebbles $W_t$. To see that this translation yields a legal pebbling, consider the derivation rule applied to get from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$.

***Axiom download*** Suppose that we download the pebbling axiom or source axiom for a vertex $v$ with immediate predecessors $u_1, \ldots, u_{\ell'}$ (where we have $\ell' = 0$ for a source $v$). All predecessors $u_i$ not having pebbles on them at time $t-1$ get white pebbles. Then $v$ gets a black pebble, if it is not already pebbled. Note that this is a legal pebble placement since all immediate predecessors of $v$ (if any) have pebbles at this point. We remark that to black-pebble $v$, we might have to remove a white pebble from $v$ first, but since all immediate predecessors have pebbles on them this poses no problems. Also, downloading the sink axiom places a white pebble on the sink $z$ if this vertex is empty, which is a legal pebbling move. By the bound on the indegree, this step involves placing at most $\ell + 1$ pebbles.

***Inference*** In this case $\mathit{Vars}(\mathbb{C}_{t-1}) = \mathit{Vars}(\mathbb{C}_t)$, so nothing happens.

***Erasure*** Suppose that the clause erased in $C$. Just apply the translation function. Suppose that this results in a pebble on $x$ disappearing. Then we have $x \in \mathit{Vars}(C)$ but $x \notin \mathit{Vars}(\mathbb{C}_t)$. Before being erased, $C$ has been resolved with some other clause (recall that $\pi$ is frugal). But as long as we did not resolve over the variable $x$, we will still have $x \in \mathit{Vars}(\mathbb{C}_t)$, and hence $\mathbb{C}$ must have been resolved over $x$ at some time $t' < t$. At this time $x$ appeared both positively and negatively in $\mathbb{C}_{t'}$, and in view of how we defined the translation from clauses to pebbles, this means that the vertex $x$ has contained a black pebble in the interval $[t', t-1]$. Thus the pebble disappearing at time $t$ is black, and black pebbles can always be removed freely.

To conclude the proof, note that during the course of the refutation all axioms must have been downloaded at least once, since $\mathit{Peb}_G$ is easily seen to be minimally unsatisfiable. In particular, this means that the sink $z$ is black-pebbled at some time during the proof, and we can decide to keep the black pebble on $z$ from that moment onwards. (This potentially adds one pebble extra to the pebbling space, but this is fine since the inequality in Theorem 3.5 is strict so there is margin for this.)

Since every time an axiom is downloaded it must also be erased at some later time, we get the time bound of $(\ell + 1)$ times the number of axiom downloads (and in fact it is easy to see that this bound can be improved by taking into account the inference steps, when nothing happens in the pebbling). The lemma follows. □

As was discussed above, Lemma 4.3 completes the proof of Theorem 4.2.

## 4.3 Obtaining Resolution Trade-offs from Pebbling

Combining Theorems 4.1 and 4.2, we can now prove that if we can find DAGs $G$ with appropriate pebbling trade-off properties, such DAGs immediately yield trade-off results in resolution. And as we will see in Section 5, there are (explicitly constructible) DAGs with the needed properties.

In order not to clutter the statement of the next theorem, we assume that the arity $d$ of the Boolean function $f$ and the indegree $\ell$ of the DAG are fixed, so that any dependence on $d$ and $\ell$ can be hidden in the asymptotical notation. (This is not much of a restriction since we will have $d = \ell = 2$ in the applications that we care about.)

**Theorem 4.4.** *Let $d$ and $\ell$ be universal constants, and let $f$ be some universally fixed non-authoritarian Boolean function of arity $d$. Suppose that $G$ is a DAG with $n$ vertices, unique sink $z$, and bounded indegree $\ell$, and that $g, h : \mathbb{N}^+ \mapsto \mathbb{N}^+$ are functions satisfying the following properties:*

- *For every $s \geq \mathsf{Peb}(G)$ there is a complete black pebbling $\mathcal{P}$ of $G$ with $\mathsf{space}(\mathcal{P}) \leq s$ and $\mathsf{time}(\mathcal{P}) \leq g(s)$.*
- *For every $s \geq \mathsf{BW\text{-}Peb}(G)$ and every complete black-white pebbling $\mathcal{P}$ of $G$ with $\mathsf{space}(\mathcal{P}) \leq s$ it holds that $\mathsf{time}(\mathcal{P}) \geq h(s)$.*

*Then the following holds for $Peb_G[f]$:*

1. *$Peb_G[f]$ is a $k$-CNF formula for some fixed $k = k(d, \ell, f)$ and has size $\mathrm{O}(n)$.*
2. *$Peb_G[f]$ is refutable in length $L(Peb_G[f] \vdash 0) = \mathrm{O}(n)$ and width $W(Peb_G[f] \vdash 0) = \mathrm{O}(1)$ simultaneously, and is also refutable in variable space $VarSp(Peb_G[f] \vdash 0) = \mathrm{O}\big(\mathsf{Peb}(G)\big)$.*
3. *For every $s \geq \mathsf{Peb}(G)$ there is a resolution refutation $\pi_s : Peb_G[f] \vdash 0$ in length $L(\pi_s) = \mathrm{O}(g(s))$ and variable space $VarSp(\pi_s) = \mathrm{O}(s)$.*
4. *The clause space of any resolution refutation is lower-bounded by $Sp(Peb_G[f] \vdash 0) \geq \mathsf{BW\text{-}Peb}(G)$, and for every $s \geq \mathsf{BW\text{-}Peb}(G)$ and every refutation $\pi_s : Peb_G[f] \vdash 0$ in clause space $Sp(\pi_s) \leq s$, it holds that $L(\pi_s) = \Omega(h(s))$.*

*All hidden constants in the asymptotical notation depend only on $d$, $\ell$, and $f$, and are independent of $G$.*

*Proof.* Item 1 is an easy consequence of Definition 3.2. Items 2 and 3 both follow from Theorem 4.1 (to get item 2, consider the trivial pebbling that black-pebbles all vertices of $G$ in topological order). Finally, Theorem 4.2 yields item 4. ☐

This theorem will be of particular interest when we can find graph families $\{G_n\}_{n=1}^{\infty}$ with $\mathsf{Peb}(G_n) = \Theta\big(\mathsf{BW\text{-}Peb}(G_n)\big)$ having trade-off functions $g_n(s) = \Theta(h_n(s))$. For such families of DAGs, Theorem 4.4 yields asymptotically tight trade-offs in resolution. We stress again that *these trade-offs hold for both clause space and variable space simultaneously* with respect to length, since the upper bounds are in terms of variable space and the lower bounds in terms of clause space.

## 5 Some Old and New Pebbling Results

Having come this far in the paper, we know that if we can find graphs with trade-off results for black-white pebbling and matching upper bounds for black pebbling, we can construct CNF formulas from these graphs with similar time-space trade-off properties in resolution. And indeed, as we show in this section, we can find graphs satisfying these properties (or in one case graphs that come sufficiently close for us to be able to get the desired result via some extra work).

First, we present some auxiliary definitions, notation and terminology in Section 5.1. Then, in Section 5.2, we prove a strong trade-off result for a very simple but surprisingly versatile family of graphs. Our results build on [CS80, CS82] and extend the results there from black-only to black-white pebbling. Finally, in Section 5.3 we review a number of results from [LT82] that will also enable us to get strong trade-offs in resolution.

We remark that all the pebbling trade-off results presented in this section are for explicitly constructible graphs.

## 5.1 Pebbling Preliminaries

We will use the following notational conventions:

- $n$ denotes the size (i.e., the number of vertices) of a DAG, or, in some cases where it is more convenient, the size to within a (small) constant factor.
- $\ell$ denotes the maximal indegree of a DAG.
- $s$ denotes pebbling space (although $s_1, s_2, \ldots$ will sometimes denote source vertices of DAGs).
- $S(G)$ denotes the source vertices of $G$ and $Z(G)$ denotes the sink vertices of $G$.

We say that the pebbling move *at time* $\sigma$ is the move resulting in the pebble configuration $\mathbb{P}_\sigma$.

### 5.1.1 Technical Definitions and Some Observations

We need to generalize our definition of pebbling slightly to distinguish somewhat different variants of pebblings and also to allow pebblings of graphs with more than one sink.

**Definition 5.1 (Conditional, persistent and visiting pebblings).** Suppose that $G$ is a DAG with sources $S$ and sinks $Z$ (one or many). Let the pebble game rules be as in Definition 2.10, and define pebbling space in the same way.

We say that a pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is *conditional* if $\mathbb{P}_0 \neq (\emptyset, \emptyset)$ and *unconditional* otherwise. Note that complete pebblings, or pebbling strategies, are always unconditional.

A complete black-white pebbling *visiting* $Z$ is a pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = \mathbb{P}_\tau = (\emptyset, \emptyset)$ and such that for every $z \in Z$, there exists a time $t_z \in [\tau]$ such that $z \in B_{t_z} \cup W_{t_z}$. The minimum space of such a visiting pebbling is denoted $\mathit{BW\text{-}Peb}^{\emptyset}(G)$, and for black pebbling we have the measure $\mathit{Peb}^{\emptyset}(G)$.

A *persistent* pebbling of $G$ is a pebbling $\mathcal{P}$ such that $\mathbb{P}_\tau = (Z, \emptyset)$. The minimum space of any complete persistent black-white or black-only pebbling of $G$ is denoted $\mathit{BW\text{-}Peb}^{z}(G)$ and $\mathit{Peb}^{z}(G)$, respectively.

That is, a visiting pebbling touches all sinks but leaves the graph empty at time $\tau$, whereas a persistent pebbling leaves black pebbles on all sinks at the end of the pebbling. If $G$ is a DAG with $m$ sinks, then it clearly holds that $\mathit{BW\text{-}Peb}^{z}(G) \leq \mathit{BW\text{-}Peb}^{\emptyset}(G) + m$ and $\mathit{Peb}^{z}(G) \leq \mathit{Peb}^{\emptyset}(G) + m$.

Intuitively, the pebblings that seem most natural and interesting are persistent pebblings of DAGs with a single sink. In our proofs, however, we will mostly be focusing on visiting pebblings. The reason that visiting pebblings will show up over and over again is that the graphs of interest will often be constructed in terms of smaller subgraph components with useful pebbling properties, and that for such subgraphs we have the following fact.

**Observation 5.2.** *Suppose that $G$ is a DAG and that $\mathcal{P}$ is any complete pebbling of $G$. Let $U \subseteq V(G)$ be any subset of vertices of $G$ and let $H = H(G, U)$ denote the induced subgraph with vertices $V(H) = U$ and edges $E(H) = \{(u, v) \in E(G) \mid u, v \in U\}$. Then the pebbling $\mathcal{P}$ restricted to the vertices in $U$ is a complete visiting pebbling strategy for $H$.*

*Proof.* It is easy to verify that if we only perform those pebbling moves in $\mathcal{P}$ that pertain to vertices in $U$, then these moves constitute a legal pebbling on $H$. Moreover, any complete pebbling of $G$ must pebble all vertices in $G$, so $\mathcal{P}$ restricted to $U$ will pebble all vertices in $H$ including the sinks of $H$. ☐

To get trade-offs in resolution for minimally unsatisfiable $k$-CNF formulas, we want DAGs with unique sinks. Most pebbling results in Section 5 are more natural to state and prove for DAGs with multiple sinks, however, but this small technicality is easily taken care of. We do this next.

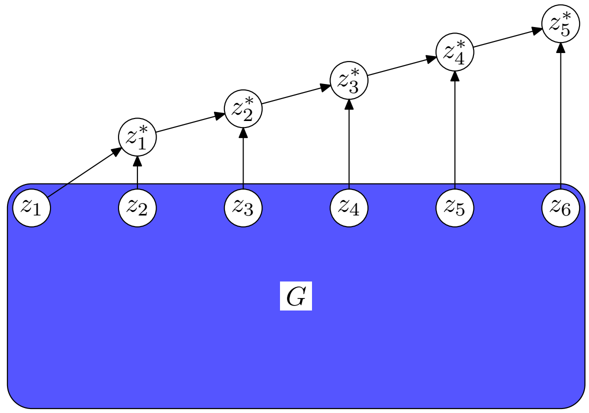


**Figure 1:** Schematic illustration of single-sink version $\widehat{G}$ of graph $G$.

**Definition 5.3 (Single-sink version).** Let $G$ be a DAG with sinks $Z(G) = \{z_1, \ldots, z_m\}$ for $m > 1$. The *single-sink version* $\widehat{G}$ of $G$ consists of all vertices and edges in $G$ plus the extra vertices $z_1^*, \ldots, z_{m-1}^*$ and the edges $(z_1, z_1^*)$, $(z_2, z_1^*)$, $(z_1^*, z_2^*)$, $(z_3, z_2^*)$, $(z_2^*, z_3^*)$, $(z_4, z_3^*)$, et cetera up to $(z_{m-2}^*, z_{m-1}^*)$, $(z_m, z_{m-1}^*)$.

That is, $\widehat{G}$ consists of $G$ with a binary tree of minimal size added on top of the sinks. See Figure 1 for a small example. The following observation is immediate.

**Observation 5.4.** *Let $G$ be a DAG with sinks $Z(G) = \{z_1, \ldots, z_m\}$ for $m > 1$. Then for any flavour of pebbling (visiting or persistent) it holds that $\mathsf{BW\text{-}Peb}\big(\widehat{G}\big) \leq \mathsf{BW\text{-}Peb}(G) + 1$ and $\mathsf{Peb}\big(\widehat{G}\big) \leq \mathsf{Peb}(G) + 1$. Moreover, for any ordering of the sinks $z_1, \ldots, z_m$ there is a pebbling strategy $\mathcal{P}$ (visiting or persistent) for $G$ in space $s$ that pebbles the sink in this order, then there is a pebbling strategy $\widehat{\mathcal{P}}$ of the same type for $\widehat{G}$ with $\mathsf{time}\big(\widehat{\mathcal{P}}\big) \leq \mathsf{time}(\mathcal{P}) + 2m$ and $\mathsf{space}\big(\widehat{\mathcal{P}}\big) \leq \mathsf{space}(\mathcal{P}) + 1$.*

To simplify the proofs of our lower bounds, we want the pebblings under consideration not to perform any obviously redundant moves. The following definition is a generalization of [GLT80] from black-only to black-white pebbling. (We are not aware of this generalization having appeared in the literature before.)

**Definition 5.5 (Frugal pebbling).** Let $\mathcal{P}$ be a complete pebbling of a DAG $G$. To every pebble placement on a vertex $v$ at time $\sigma$ we associate the *pebbling interval* $[\sigma, \tau)$, where $\tau$ is the first time after $\sigma$ when the pebble is removed from $v$ again (or $\tau = \infty$, say, if this never happens).

If a sink $z_i \in Z(G)$ is pebbled for the first time at time $\sigma$, then the pebble on $z_i$ is *essential* during the pebbling interval $[\sigma, \tau)$. A pebble on a non-sink vertex $v$ is essential during $[\sigma, \tau)$ if either an essential black pebble is placed on an immediate successor of $v$ during $(\sigma, \tau)$ or an essential white pebble is removed from an immediate successor of $v$ during $(\sigma, \tau)$. Any other pebble placements on any vertices are non-essential.

The pebbling strategy $\mathcal{P}$ is *frugal* if all pebbles in $\mathcal{P}$ are essential at all times.

Without loss of generality, we can assume that all pebblings we deal with are frugal.

**Lemma 5.6.** *For any complete pebbling $\mathcal{P}$ (black or black-white, visiting or persistent) there is a frugal pebbling $\mathcal{P}'$ of the same type such that $\mathsf{time}(\mathcal{P}') \leq \mathsf{time}(\mathcal{P})$ and $\mathsf{space}(\mathcal{P}') \leq \mathsf{space}(\mathcal{P})$.*

*Proof.* Just delete any non-essential pebbles from $\mathcal{P}$ and verify that what remains is a legal pebbling. ☐

One minor technical snag is that we will need to assume not only that complete pebblings are frugal, but that this also holds for *conditional pebblings* (Definition 5.1). This is no real problem, however, since we can always assume that the conditional pebblings we are dealing with are subpebblings of some larger, unconditional pebbling. More formally, we can define a conditional pebbling to be frugal if it satisfies the condition in Definition 5.5 that any pebble placed on a non-sink vertex $v$ stays until either a black pebble is placed on an immediate successor of $v$ or a white pebble is removed from an immediate successor of $v$.

### 5.1.2 Some Upper and Lower Bounds

If we do not care about space, the easiest way to pebble a DAG is to place black pebbles on the vertices in topological order (and then remove all pebbles from non-sink vertices). Since we will have reason to use this pebbling strategy on occasion in what follows, we give it a name for reference.

**Observation 5.7 (Trivial pebbling).** *Any DAG $G$ can be completely, persistently black-pebbled in space at most $|V(G)|$ and time at most $2 \cdot |V(G)|$ simultaneously.*

Another easy upper bound on pebbling price can be given in terms of the fan-in and depth of a DAG.

**Definition 5.8 (Depth).** The *depth* of a DAG $G$ is the length of a longest path from a source to a sink in $G$.

**Observation 5.9.** *Any DAG $G$ with maximal indegree $\ell$ and depth $d$ has a black pebbling strategy in space at most $d\ell + 1$.*

*Proof.* By induction over the depth. The base case is immediate. For a graph of depth $d+1$, pebble the sinks one by one. For each sink we can pebble its immediate predecessors with $d\ell + 1$ pebbles each by induction. Placing black pebbles on the immediate predecessors one by one and leaving them there, we never use more than $(d\ell + 1) + (\ell - 1)$ pebbles simultaneously. Finally, keeping the at most $\ell$ pebbles on the predecessors, pebble the sink. ☐

Next follows a simple but important lemma that is central to most black-white pebbling lower bounds.

**Lemma 5.10 ([GT78]).** *Suppose that $Q : u \rightsquigarrow v$ is a path in $G$ and that $\mathcal{P} = \{\mathbb{P}_\sigma, \mathbb{P}_{\sigma+1}, \ldots, \mathbb{P}_\tau\}$ is a pebbling such that the whole path $Q$ is completely free of pebbles at times $\sigma$ and $\tau$ but the endpoint $v$ is pebbled at some point in the time interval $(\sigma, \tau)$. Then the starting point $u$ is pebbled during $(\sigma, \tau)$ as well.*

*Proof.* By induction over the length of the path $Q$. The base case $u = v$ is trivial. For the induction step, let $w$ be the immediate successor of $u$ on $Q$. By the induction hypothesis, $w$ is pebbled and unpebbled again sometime during $(\sigma, \tau)$. Then $u$ must be covered by a pebble either when the pebble on $w$ is placed there (if this pebble is black) or when it is removed (if it is white). The lemma follows. ☐

A common graph in many pebbling constructions is the *pyramid* (see Figure 2 for an example), the pebbling price of which is well understood.

**Definition 5.11 (Pyramid graph).** The *pyramid graph* $\Pi_h$ of height $h \geq 1$ is a layered DAG with $h + 1$ levels, where there is one vertex on the highest level (the sink $z$), two vertices on the next level et cetera down to $h + 1$ vertices at the lowest level $0$. The $i$th vertex at level $L$ has incoming edges from the $i$th and $(i + 1)$st vertices at level $L - 1$.

**Theorem 5.12.** *The black pebbling price of a pyramid of height $h$ is $\mathsf{Peb}(\Pi_h) = h+2$ and there is a linear-time pebbling achieving this bound. The black-white pebbling price is $\mathsf{BW\text{-}Peb}^{\emptyset}(\Pi_h) = h/2 + \mathrm{O}(1)$, and for even height there is the exact bound $\mathsf{BW\text{-}Peb}^{\emptyset}(\Pi_{2h}) = h + 2$.*

*Proof sketch.* The lower bound for black pebbling is from Cook [Coo74], and it is easy to construct a linear-time pebbling matching this bound by pebbling the pyramid bottom-up, layer by layer.

The black-white pebbling strategy for pyramids in space $h/2 + \mathrm{O}(1)$ can be obtained from the strategy for trees in Lengauer and Tarjan [LT80], and Klawe [Kla85] showed that $h/2 + \mathrm{O}(1)$ is also a lower bound. The exact bound for pyramids of even height can be found in the exposition of Klawe's proof in [Nor09]. ☐

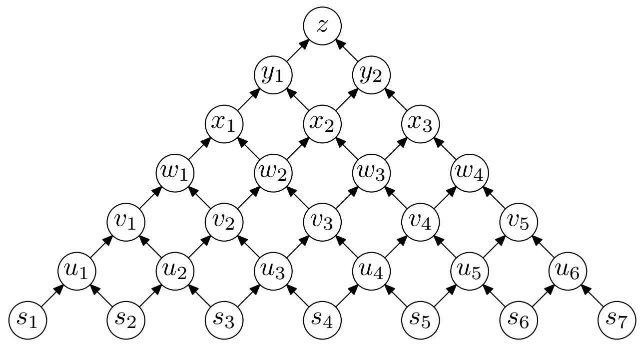


**Figure 2:** Pyramid $\Pi_6$ of height $6$.

Another important building block in many pebbling results are so-called *superconcentrators*.

**Definition 5.13 (Superconcentrator).** A directed acyclic graph $G$ is an $N$*-superconcentrator* if it has $N$ sources $S = \{s_1, \ldots, s_N\}$, $N$ sinks $Z = \{z_1, \ldots, z_N\}$, and for any subsets $S'$ and $Z'$ of sources and sinks of size $|S'| = |Z'| = k$ it holds that there are $k$ vertex-disjoint paths between $S'$ and $Z'$ in $G$.

For our pebbling purposes, we will be interested in superconcentrators with number of vertices and edges linear in $N$ (in addition, we will want them to have bounded indegree, but this extra requirement is easy to take care of). Valiant [Val76] proved the existence of such graphs, and Gabber and Galil [GG81] provided the first explicit construction based on an earlier non-explicit one by Pippenger [Pip77]. We remark that the superconcentrators in [GG81] have logarithmic depth. The currently best known construction (i.e., with lowest edges-to-vertices ratio) that we are aware of is due to Alon and Capalbo [AC03].

Here is an important lemma that explains why superconcentrators are good building blocks if we want to construct graphs that are hard to pebble.

**Lemma 5.14 ([LT82]).** *Suppose that $G$ is an $N$-superconcentrator and that $\mathcal{P} = \{\mathbb{P}_\sigma, \mathbb{P}_{\sigma+1}, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling such that $\mathsf{space}(\mathbb{P}_\sigma) \leq s_\sigma$, $\mathsf{space}(\mathbb{P}_\tau) \leq s_\tau$, and $\mathcal{P}$ pebbles at least $s_\sigma + s_\tau + 1$ sinks during the closed time interval $[\sigma, \tau]$. Then $\mathcal{P}$ pebbles and unpebbles at least $N - s_\sigma - s_\tau$ different sources during the open time interval $(\sigma, \tau)$.*

*Proof.* Suppose not. Then $\mathcal{P}$ pebbles some set of $s_\sigma + s_\tau + 1$ sinks without pebbling some set of $s_\sigma + s_\tau + 1$ sources. Fix such sets of sources and sink vertices and consider the vertex-disjoint paths from sources to sinks. Then at least one path is empty both at time $\sigma$ and at time $\tau$ and the end point of the path is pebbled during the interval $(\sigma, \tau)$ but not the starting point. This contradicts Lemma 5.10. ☐

We immediately get the following corollary.

**Corollary 5.15 ([LT82]).** *Any complete black-white pebbling of an $N$-superconcentrator in space at most $s$ has to pebble at least $\Omega\bigl(N^2/s\bigr)$ sources (so, in particular, this is a lower bound on the pebbling time).*

## 5.2 A New Pebbling Trade-off Result

In this section we present the third main contribution of this paper, which is a graph family that provides us with a number of interesting time-space trade-offs for different parameter settings. These trade-offs have the property that the lower bounds are in terms of black-white pebbling while the upper bounds are in terms of black-only pebbling, and thanks to this we can apply the machinery of Theorem 3.5 on page 14 and Theorem 4.4 on page 26 on these graphs to derive corresponding trade-offs in proof complexity for resolution.

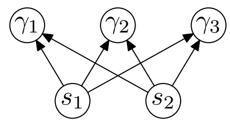


**Figure 3:** Base case $\Gamma(3, 1)$ for Carlson-Savage graph with $3$ spines and sinks.

### 5.2.1 Definition of Graph Family and Statement of Trade-off

Our graph family is built on a construction by Carlson and Savage [CS80, CS82]. Carlson and Savage only prove their trade-off for black pebbling, however, and in order to get results for black-white pebbling we have to modify the construction somewhat and also apply some new ideas in the proofs. The next definition will hopefully be easier to parse if the reader first studies the illustrations in Figures 3 and 4.

**Definition 5.16 (Carlson-Savage graph).** We define a two-parameter graph family $\Gamma(c, r)$, for $c, r \in \mathbb{N}^+$, by induction over $r$. The base case $\Gamma(c, 1)$ is a DAG consisting of two sources $s_1, s_2$ and $c$ sinks $\gamma_1, \ldots, \gamma_c$ with directed edges $(s_i, \gamma_j)$, for $i = 1, 2$ and $j = 1, \ldots, c$, i.e., edges from both sources to all sinks. The graph $\Gamma(c, r + 1)$ is a DAG with $c$ sinks which is built from the following components:

- $c$ disjoint copies $\Pi^1_{2r}, \ldots, \Pi^c_{2r}$ of a pyramid (Definition 5.11) of height $2r$, where we let $z_1, \ldots, z_c$ denote the pyramid sinks.
- one copy of $\Gamma(c, r)$, for which we denote the sinks by $\gamma_1, \ldots, \gamma_c$.
- $c$ disjoint and identical *spines*, where each spine is composed of $cr$ *sections*, and every section contains $2c$ vertices. We let the vertices in the $i$th section of a spine be denoted $v[i]_1, \ldots, v[i]_{2c}$.

The edges in $\Gamma(c, r + 1)$ are as follows:

- All "internal edges" in $\Pi^1_{2r}, \ldots, \Pi^c_{2r}$ and $\Gamma(c, r)$ are present also in $\Gamma(c, r + 1)$.
- For each spine, there are edges $\bigl(v[i]_j, v[i]_{j+1}\bigr)$ for all $j = 1, \ldots, 2c - 1$ within each section $i$ and edges $\bigl(v[i]_{2c}, v[i + 1]_1\bigr)$ from the end of a section to the beginning of next for $i = 1, \ldots, cr - 1$, i.e., for all sections but the final one, where $v[cr]_{2c}$ is a sink.
- For each section $i$ in each spine, there are edges $\bigl(z_j, v[i]_j\bigr)$ from the $j$th pyramid sink to the $j$th vertex in the section for $j = 1, \ldots, c$, as well as edges $\bigl(\gamma_j, v[i]_{c+j}\bigr)$ from the $j$th sink in $\Gamma(c, r)$ to the $(c + j)$th vertex in the section for $j = 1, \ldots, c$.

We now make the formal statements of the trade-off properties that these DAGs possess. The proofs of all the statements are postponed to Section 5.2.2. First, we collect some basic properties.

**Lemma 5.17.** *The graphs* $\Gamma(c, r)$ *are of size* $\bigl|V\bigl(\Gamma(c, r)\bigr)\bigr| = \Theta\bigl(cr^3 + c^3r^2\bigr)$*, and have black-white pebbling price* $\mathit{BW\text{-}Peb}^{\emptyset}\bigl(\Gamma(c, r)\bigr) = r + 2$ *and black pebbling price* $\mathit{Peb}^{\emptyset}\bigl(\Gamma(c, r)\bigr) = 2r + 1$.

This tells us that the minimum pebbling space required grows linearly with the recursion depth $r$ but is independent of the number of spines $c$ of the DAG.

Next, we show that there is a linear-time completely black pebbling of $\Gamma(c, r)$ in space linear in the sum of the parameters. This is in fact a strict improvement (though easily obtained) of the corresponding result in [CS80, CS82].

**Lemma 5.18.** *The graph* $\Gamma(c, r)$ *has a persistent black pebbling strategy* $\mathcal{P}$ *in time linear in the size of the DAG and with space* $\mathrm{O}(c + r)$.

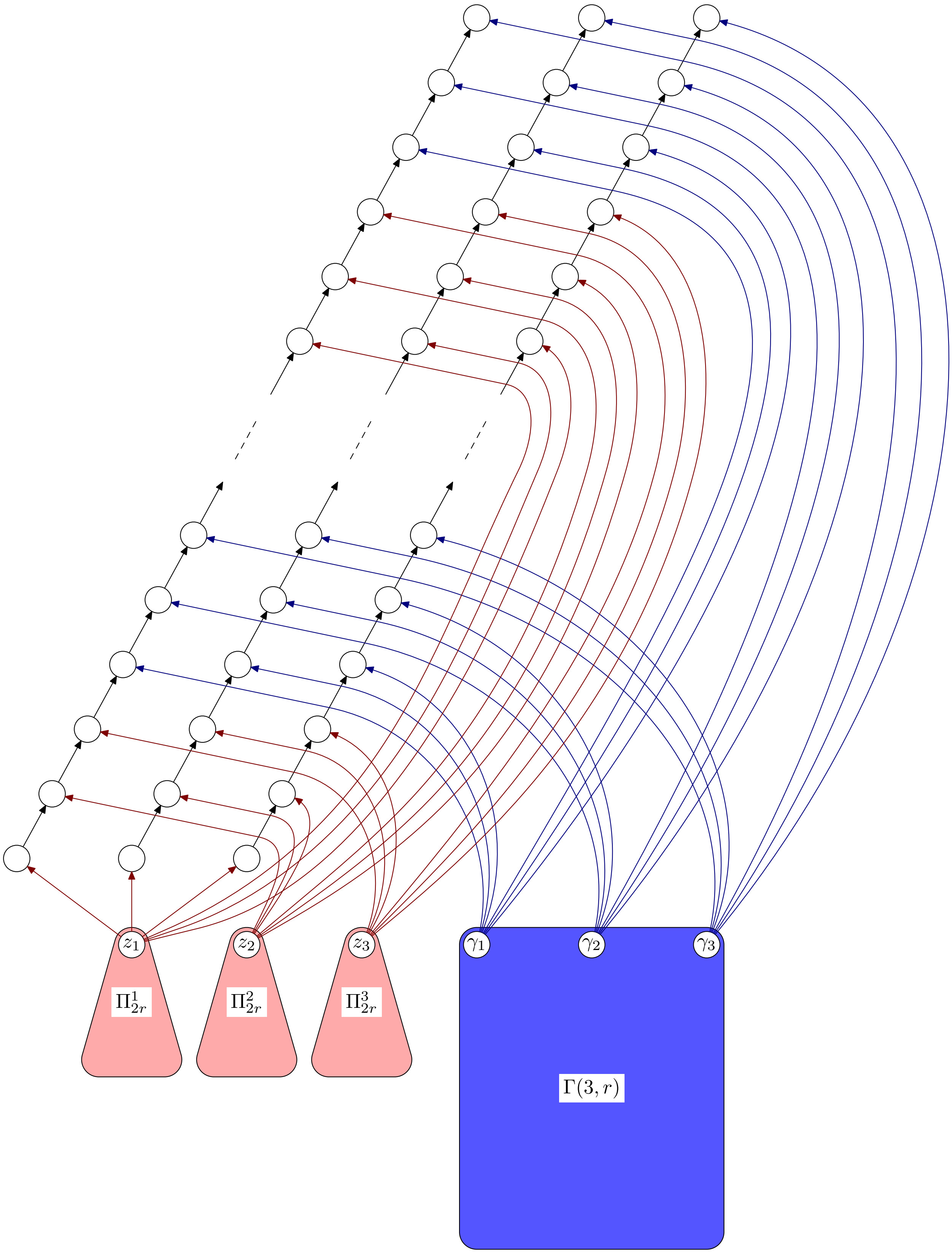


**Figure 4:** Inductive definition of Carlson-Savage graph $\Gamma(3, r + 1)$ with $3$ spines and sinks.

The proof is by induction, and the idea in the induction step for $\Gamma(c, r+1)$ is to make a persistent pebbling of $\Gamma(c,r)$ in space $\mathrm{O}(c+r)$, then pebble the pyramids $\Pi^1_{2r}, \ldots, \Pi^c_{2r}$ one by one in linear time and space $\mathrm{O}(r)$, and finally, using the $2c$ black pebbles on $z_1, \ldots, z_c, \gamma_1, \ldots, \gamma_c$ that we have left in place, to pebble all $c$ spines in parallel with $\mathrm{O}(c)$ extra pebbles.

The main result of this section is the following theorem, which allows us to get a variety of pebbling trade-off results if we choose the parameters $c$ and $r$ appropriately.

**Theorem 5.19.** *Suppose that $\mathcal{P}$ is a complete visiting black-white pebbling of $\Gamma(c,r)$ with*

$$\mathsf{space}(\mathcal{P}) < \mathsf{BW\text{-}Peb}^{\emptyset}\bigl(\Gamma(c,r)\bigr) + s = (r+2) + s$$

*for $0 < s \leq c/8 - 1$. Then the time required to perform $\mathcal{P}$ is lower-bounded by*

$$\mathsf{time}(\mathcal{P}) \geq \left(\frac{c-2s}{4s+4}\right)^r \cdot r! \ .$$

As has already been noted, we defer the proof of Theorem 5.19 to Section 5.2.2, but let us nevertheless try to provide some intuition as to why the theorem should be true.

For simplicity, let us focus on black-only pebbling strategies. Inductively, suppose that the trade-off in Theorem 5.19 has been proven for $\Gamma(c,r)$ and consider $\Gamma(c,r+1)$. Any pebbling strategy for this DAG will have to pebble through all sections in all spines. Consider the first section anywhere, let us say on spine $j$, that has been completely pebbled, i.e., there have been pebbles placed on and removed from all vertices in the section. Let us say that this happens at time $\tau_1$. But this means that $\Gamma(c,r)$ and all pyramids $\Pi^1_{2r}, \ldots, \Pi^c_{2r}$ must have been completely pebbled during this part of the pebbling as well. Fix any pyramid and consider some point in time $\sigma_1 < \tau_1$ when the number of pebbles in this pyramid reaches the space $r + \mathrm{O}(1)$ required by the known lower bound on pyramid pebbling price. At this point, the rest of the graph must contain very few pebbles. In particular, there are very few pebbles on the subgraph $\Gamma(c,r)$ at time $\sigma_1$, so we can think of $\Gamma(c,r)$ as being completely empty of pebbles for all practical purposes.

Let us now shift the focus to the next section in the spine $j$ that is completed, say, at time $\tau_2 > \tau_1$. Again, we can argue that some pyramid is completely pebbled in the time interval $[\tau_1, \tau_2]$, and thus has $r + \mathrm{O}(1)$ pebbles on it at some time $\sigma_2 > \tau_1 > \sigma_1$. This means that we can think of $\Gamma(c,r)$ as being completely empty at time $\sigma_2$ as well.

But note that all sinks in the subgraph $\Gamma(c,r)$ must have been pebbled in the time interval $[\sigma_1, \sigma_2]$, and since we know that $\Gamma(c,r)$ is (almost) empty at times $\sigma_1$ and $\sigma_2$, this allows us to apply the induction hypothesis. Since $\mathcal{P}$ has to pebble through a lot of sections in different spines, we will be able to repeat the above argument many times and apply the induction hypothesis on $\Gamma(c,r)$ in each round. Adding up all the lower bounds obtained in this way, we see that the induction step goes through.

This is essentially the proof in [CS80, CS82] for black pebbling, modulo a number of technical details that we glossed over. For black-white pebbling, these technical complications grow more serious. The main problem is that in contrast to a black pebbling that has to proceed through the DAG in some kind of bottom-up fashion, a black-white pebbling can place and remove pebbles anywhere in the DAG at any time. Therefore, it is more difficult to control the progress of a black-white pebbling, and we have to work harder in the proof of our theorem.

Also, it should be noted that the added complications when going from black to black-white pebbling result in our bounds for black-white pebbling being slightly worse than the ones in [CS80, CS82] for black pebbling only. More specifically, Carlson and Savage are able to prove their results for DAGs having only $\Theta(r)$ sections per spine, whereas we need $\Theta(cr)$ sections. This blows up the number of vertices, which in turn weakens the trade-offs measured in terms of graph size.

It would be interesting to find out whether our proof, presented below, could in fact be made to work for graphs with only $\mathrm{O}(r)$ sections per spine. If so, this would immediately improve all the trade-off results for resolution in Section 6 that we obtain based on the graphs in Definition 5.16.

### 5.2.2 Proofs of Lemma 5.17, Lemma 5.18, and Theorem 5.19

Before proving the results claimed in Section 5.2.1, we establish a couple of useful auxiliary lemmas. The first lemma below gives us information about how the spines in $\Gamma(c, r)$ are pebbled. We will use this information repeatedly in what follows.

**Lemma 5.20.** *Suppose that $G$ is a DAG and that $v$ is a vertex in $G$ with a path $Q$ to some sink $z_i \in Z(G)$ such that all vertices in $Q \setminus \{z_i\}$ have outdegree 1. Then any frugal black-white pebbling strategy pebbles $v$ exactly once, and the path $Q$ contains pebbles during one contiguous time interval.*

*Proof.* By induction from the sink backwards. The induction base is immediate. For the inductive step, suppose $v$ has immediate successor $w$ and that $w$ is pebbled exactly once.

If $w$ is black-pebbled at time $\sigma$, then $v$ has been pebbled before this and the first pebble placed on $v$ stays until time $\sigma$. No second placement of a pebble on $v$ after time $\sigma$ can be essential since $v$ has no other immediate successor than $w$. If $w$ is white-pebbled and the pebble is removed at time $\sigma$, then the first pebble placed on $v$ stays until time $\sigma$ and no second placement of a pebble on $v$ after time $\sigma$ can be essential.

Thus each vertex on the path is pebbled exactly once, and the time intervals when a vertex $v$ and its successor $w$ have pebbles on them overlap. The lemma follows. □

The second lemma speaks about subgraphs $H$ of a DAG $G$ whose only connection to the rest of the DAG $G \setminus H$ are via the sink of $H$. Note that the pyramids in $\Gamma(c, r)$ satisfy this condition.

**Lemma 5.21.** *Let $G$ be a DAG and $H$ a subgraph in $G$ such that $H$ has a unique sink $z_h$ and the only edges between $V(H)$ and $V(G) \setminus V(H)$ emanate from $z_h$. Suppose that $\mathcal{P}$ is any frugal complete pebbling of $G$ having the property that $H$ is completely empty of pebbles at some given time $\tau'$ but at least one vertex of $H$ has been pebbled during the time interval $[0, \tau']$. Then $\mathcal{P}$ pebbles $H$ completely during the interval $[0, \tau']$.*

*Proof.* Suppose that $v \in V(H)$ is pebbled at time $\sigma' < \tau'$. As in the proof of Lemma 5.10, we can argue by induction over the length of the longest path from $v$ to the sink $z_h$ of $H$ that $z_h$ must also be pebbled before time $\tau'$. Note that such a path exists since the sink $z_h$ is unique, and that any path starting in $v$ must hit $z_h$ sooner or later, since this vertex is the only way out of $H$ into the rest of $G$. Since $H$ is empty at times $0$ and $\tau'$, we conclude that $\mathcal{P}$ makes a complete visiting pebbling of $H$ during $[0, \tau']$. □

Let us now establish that the size and pebbling price of $\Gamma(c, r)$ are as claimed.

*Proof of Lemma 5.17.* The base case $\Gamma(c, 1)$ for the Carlson-Savage graph in Definition 5.16 has size $c + 2$. A pyramid of height $h$ has $(h+1)(h+2)/2$ vertices, so the $c$ pyramids of height $2(r-1)$ in $\Gamma(c, r)$ contribute $cr(2r-1)$ vertices. The $c$ spines with $cr$ sections of $2c$ vertices each contribute a total of $2c^3 r$ vertices. And then there are the vertices in $\Gamma(c, r-1)$. Summing up, the total number of vertices in $\Gamma(c, r)$ is

$$(c + 2) + \sum_{i=2}^{r} \bigl(ci(2i - 1) + 2c^3 i\bigr) = \Theta\bigl(cr^3 + c^3 r^2\bigr) \tag{33}$$

as is stated in the lemma.

Clearly, *BW-Peb*$^{\emptyset}(\Gamma(c, 1)) = $ *Peb*$^{\emptyset}(\Gamma(c, 1)) = 3$, since pebbling a vertex with fan-in 2 requires 3 pebbles and $\Gamma(c, 1)$ can be completely pebbled in this way by placing pebbles on the two sources and then pebble and unpebble the sinks one by one.

Suppose inductively that *BW-Peb*$^{\emptyset}(\Gamma(c, r)) = r + 2$ and consider $\Gamma(c, r + 1)$. It is straightforward to see that *BW-Peb*$^{\emptyset}(\Gamma(c, r + 1)) \leq r + 3$. Every pyramid $\Pi^{j}_{2r}$ can be completely pebbled with $r + 2$ pebbles (Theorem 5.12). We can pebble each spine bottom-up in the following way, section by section. Suppose by induction that we have a black pebble on the last vertex $v[i-1]_{2c}$ in the $(i-1)$st section. Keeping the pebble on $v[i-1]_{2c}$, perform a complete visiting pebbling of $\Pi^1_{2r}$. At some point during this pebbling we must have a pebble on the pyramid sink $z_1$ and at most $r$ other pebbles on the pyramid (simply because without loss of generality some pebble placement on $z_1$ must be followed by a removal or placement of a pebble on some other vertex). At this time, place a black pebble on $v[i]_1$ and remove the pebble from $v[i-1]_{2c}$. Complete the pebbling of $\Pi^1_{2r}$, leaving the pyramid empty. Performing complete visiting pebblings of $\Pi^2_{2r}, \ldots, \Pi^c_{2r}$ in the same way allows us to move the black pebble along $v[i]_2, \ldots, v[i]_c$, never exceeding total pebbling space $r+3$. It is easy to see that in the same way, for every visiting pebbling $\mathcal{P}$ of $\Gamma(c,r)$ there must exist times $\sigma_i$ for all $i = 1, \ldots, c$, when $\mathsf{space}(\mathbb{P}_{\sigma_i}) < \mathsf{space}(\mathcal{P})$ and the sink $\gamma_i$ contains a pebble. Performing a minimum-space pebbling of $\Gamma(c,r)$, possibly $c$ times if necessary, this allows us to advance the black pebble along $v[i]_{c+1}, \ldots, v[i]_{2c}$, never exceeding total pebbling space $r+3$. This show that $\Gamma(c, r+1)$ can be completely pebbled with $r+3$ pebbles. A simple pattern-matching of this argument for black pebbling (appealing to Theorem 5.12 for the black pebbling price of pyramids) also yields $\mathsf{Peb}^{\emptyset}(\Gamma(c,r)) \leq 2r+3$.

To prove that there are matching lower bounds for the pebbling constructed above, it is sufficient to show that some pyramid $\Pi^j_{2r}$ must be completely pebbled while there is at least one pebble on $\Gamma(c, r+1)$ outside of $\Pi^j_{2r}$. To see why, note that if we can prove this, then simply by using the the fact that $\mathsf{BW\text{-}Peb}^{\emptyset}(\Pi_{2r}) = r+2$ and $\mathsf{BW\text{-}Peb}^{\emptyset}(\Pi_{2r}) = 2r+2$ and adding an additive constant 1 for the pebble outside of $\Pi^j_{2r}$ we have the matching lower bounds that we need. We present the argument for black-white pebbling, which is the harder case. The black-only pebbling case is handled completely analogously.

Suppose to get a contradiction that there is a complete visiting pebbling strategy $\mathcal{P}$ for $\Gamma(c, r+1)$ in space $r+2$. By Observation 5.2, $\mathcal{P}$ performs a complete visiting pebbling of every pyramid $\Pi^j_{2r}$. Consider the first time $\tau_1$ when some pyramid has been completely pebbled and let this pyramid be $\Pi^{j_1}_{2r}$. Then at some time $\sigma_1 < \tau_1$ there are $r+2$ pebbles on $\Pi^{j_1}_{2r}$ and the rest of the graph $\Gamma(c, r+1)$ must be empty of pebbles at this point.

We claim that this implies that no vertex in $\Gamma(c, r+1)$ outside of the pyramid $\Pi^{j_1}_{2r}$ has been pebbled before time $\sigma_1$. Let us prove this crucial fact by a case analysis.

1. No vertex $v$ in any other pyramid $\Pi^{j'}_{2r}$ can have been pebbled before time $\sigma_1$. For if so, Lemma 5.21 says that $\Pi^{j'}_{2r}$ has been completely pebbled before time $\sigma_1$, contradicting that $\Pi^{j_1}_{2r}$ is the first pyramid completely pebbled by $\mathcal{P}$.

2. No vertex on a spine has been pebbled before time $\sigma_1$. This is so since Lemma 5.20 tells us that if some vertex on a spine has been pebbled, then the whole spine must have been pebbled in view of the fact that it is empty at time $\sigma_1$. But then Lemma 5.10 implies that all pyramid sinks must have been pebbled. This case has already been excluded.

3. Finally, no vertex $v$ in $\Gamma(c,r)$ can have been pebbled before time $\sigma_1$. Otherwise the frugality of $\mathcal{P}$ implies (by pattern matching on the arguments in the proofs of Lemmas 5.10 and 5.20) that some successor of $v$ must have been pebbled as well, and some successor of that successor et cetera, all the way up to where $\Gamma(c,r)$ connects with the spines. But we have ruled out the possibility that a spine vertex has been pebbled.

This establishes the claim, and we are now almost done. Before clinching the argument, we need to make a couple of observations. Note first that by frugality, we can conclude that at some time in the interval $(\sigma_1, \tau_1)$ some vertex in some spine must be pebbled. This is so since the pyramid sink $z_{j_1}$ has been pebbled before time $\tau_1$ and all of $\Pi^{j_1}_{2r}$ is empty at time $\tau_1$ but all spines are empty at time $\sigma_1 < \tau_1$. But then Lemma 5.20 tells us that there will remain a pebble on this spine until all of the spine has been completely pebbled.

Consider now the second pyramid $\Pi^{j_2}_{2r}$ completely pebbled by $\mathcal{P}$, say, at time $\tau_2$. At some point in time $\sigma_2 < \tau_2$ we have $r+2$ pebbles on $\Pi^{j_2}_{2r}$, and moreover $\sigma_2 > \tau_1$ since $\Pi^{j_2}_{2r}$ is empty at time $\tau_1$. But now it must hold that either there is a pebble on a spine at this point, or, if all spines are completely empty, that some spine has been completely pebbled. If some spine has been completely pebbled, however, this in turn implies (appealing to Lemma 5.10 again) that there must be some pebble somewhere in some other pyramid $\Pi^{j'}_{2r}$ at time $\sigma_2$. Thus the pebbling space exceeds $r + 2$ and we have obtained our contradiction. The lemma follows. ☐

Studying the pebbling strategies in the proof of Lemma 5.17, it is not hard to see that they are terribly inefficient. The subgraphs in $\Gamma(c, r)$ will be pebbled over and over again, and for every step in the recursion the time required multiples. We next show that if we are just a bit more generous with the pebbling space, then we can get down to linear time.

*Proof of Lemma 5.18.* We want to prove that $\Gamma(c, r)$ has a persistent black pebbling strategy $\mathcal{P}$ in linear time and in space $\mathrm{O}(c + r)$. Suppose that there is such a pebbling strategy $\mathcal{P}_r$ for $\Gamma(c, r)$. We show how to construct a pebbling $\mathcal{P}_{r+1}$ for $\Gamma(c, r + 1)$ inductively. Note that the base case for $\Gamma(c, 1)$ is trivial.

The construction of $\mathcal{P}_{r+1}$ is very straightforward. First use $\mathcal{P}_r$ to make a persistent pebbling of $\Gamma(c, r)$ in space $\mathrm{O}(c + r)$. At the end of $\mathcal{P}_r$, we have $c$ pebbles on the sinks $\gamma_1, \ldots, \gamma_c$. Keeping these pebbles in place, pebble the pyramids $\Pi^1_{2r}, \ldots, \Pi^c_{2r}$ persistently one by one in linear time and space $\mathrm{O}(r)$. We leave pebbles on all pyramid sinks $z_1, \ldots, z_c$. This stage of the pebbling only requires space $\mathrm{O}(c + r)$ and at the end we have $2c$ black pebbles on all pyramid sinks $z_1, \ldots, z_c$ and all sinks $\gamma_1, \ldots, \gamma_c$ of $\Gamma(c, r)$. Keeping all these pebbles in place, we can pebble all $c$ spines in parallel in linear time with $c + 1$ extra pebbles. ☐

It remains to prove the trade-off result in Theorem 5.19. It is clear that this theorem follows if we can prove the next, slightly stronger, statement.

**Lemma 5.22.** *Suppose that* $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ *is a conditional black-white pebbling on* $\Gamma(c, r)$ *and that* $s$ *is a constant satisfying the following properties:*

1. $0 < s \leq c/8 - 1$.
2. $\mathcal{P}$ *pebbles all sinks in* $\Gamma(c, r)$ *during the time interval* $[\sigma, \tau]$.
3. $\max\{\mathit{space}(\mathbb{P}_\sigma), \mathit{space}(\mathbb{P}_\tau)\} < s$.
4. $\mathit{space}(\mathcal{P}) < \mathit{BW\text{-}Peb}^{\emptyset}\big(\Gamma(c, r)\big) + s = (r + 2) + s$.

*Then it holds that* $\mathit{time}(\mathcal{P}) = \tau - \sigma \geq \left(\frac{c-2s}{4s+4}\right)^r \cdot r!$.

We will have to spend some time working on this lemma before the proof is complete. We start by establishing two additional auxiliary lemmas that upper-bound how many pyramids and spine sections can contain pebbles simultaneously at any one given time in a pebbling subjected to space constraints as in Lemma 5.22. The claims in the two lemmas are very similar in spirit, as are the proofs, so we state the lemmas together and then present the proofs together.

**Lemma 5.23.** *Suppose that* $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ *is a conditional black-white pebbling on* $\Gamma(c, r)$ *and that* $s$ *is a constant such that* $\mathcal{P}$ *and* $s$ *satisfy the conditions in Lemma 5.22. Then at all times during the pebbling* $\mathcal{P}$ *strictly less than* $4(s + 1)$ *pyramids* $\Pi^j_{2r}$ *contain pebbles simultaneously.*

**Lemma 5.24.** *Suppose that* $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ *is a conditional black-white pebbling on* $\Gamma(c, r)$ *and that* $s$ *is a constant such that* $\mathcal{P}$ *and* $s$ *satisfy the conditions in Lemma 5.22. Then at all times during the pebbling* $\mathcal{P}$ *strictly less than* $4(s + 1)$ *spine sections contain pebbles simultaneously.*

Note that Lemma 5.24 provides a total bound on the number of pebbled sections in all $c$ spines. There might be spines containing several sections being pebbled simultaneously (in fact, this is exactly what one would expect a black-white pebbling to do in order to optimize the time given the space constraints), but what Lemma 5.24 says is that if we fix an arbitrary time $t \in [\sigma, \tau]$, add up the number of sections containing pebbles at time $t$ in each spine, and sum over all spines, we never exceed $4(s+1)$ sections in total at any point in time $t \in [\sigma, \tau]$.

*Proof of Lemma 5.23.* Suppose that on the contrary, there is some time $t^* \in (\sigma, \tau)$ when at least $4s+4$ pyramids $\Pi^j$ in $\Gamma(c,r)$ contain pebbles. Of these pyramids, at least $2s+4$ are empty both at time $\sigma$ and at time $\tau$ since $\mathit{space}(\mathbb{P}_\sigma) < s$ and $\mathit{space}(\mathbb{P}_\tau) < s$. By Lemma 5.21, these pyramids, which we denote $\Pi^1, \ldots, \Pi^{2s+4}$, are completely pebbled. We conclude that for every $\Pi^j$, $j = 1, \ldots, 2s+4$, there is an interval $[\sigma_j, \tau_j]$ such that $t^* \in (\sigma_j, \tau_j)$ and $\Pi^j$ is empty at times $\sigma_j$ and $\tau_j$ but contains pebbles throughout the interval $(\sigma_j, \tau_j)$ during which it is completely pebbled.

For each $\Pi^j$ there must exist some time $t_j^* \in (\sigma_i, \tau_i)$ when there are at least $r+1 = \mathit{BW\text{-}Peb}^{\emptyset}(\Pi^j)$ pebbles. Fix such a time $t_j^*$ for every pyramid $\Pi^j$ and assume that all $t_j^*$, $j = 1, \ldots, 2s+4$, are sorted in increasing order. We have two possible cases:

1. At least half of all $t_j^*$ occur before (or at) time $t^*$, i.e., they satisfy $t_j^* \leq t^*$. If so, look at the largest $t_j^* \leq t^*$. At this time there are at least $r+1$ pebbles on $\Pi^j$ and at least $\frac{2s+4}{2} - 1 = s+1$ pebbles on other pyramids, which means that $\mathit{space}(\mathbb{P}_{t_j^*}) \geq (r+2)+s$. In other words, $\mathcal{P}$ exceeds the space restrictions contradicting our assumptions.

2. At least half of all $t_j^*$ occur after time $t^*$, i.e., they satisfy $t_j^* > t^*$. If we consider the smallest $t_j^*$ larger than $t^*$ we can again conclude that $\mathit{space}(\mathbb{P}_{t_j^*}) \geq (r+1)+(s+1)$, which is a contradiction.

Hence, if $\mathcal{P}$ is a pebbling that complies with the restrictions in Lemma 5.22, it can never be the case that $4s+4$ pyramids $\Pi^j$ in $\Gamma(c,r)$ contain pebbles simultaneously during $\mathcal{P}$. □

*Proof of Lemma 5.24.* Suppose in order to get a contradiction that at some time $t^* \in (\sigma, \tau)$ at least $4s+4$ sections contain pebbles. At least $2s+4$ of these sections are empty at times $\sigma$ and $\tau$. Let us denote these sections $R^1, \ldots, R^{2s+4}$. Appealing to Lemma 5.20, we conclude that there are intervals $[\sigma_j, \tau_j]$ for $j = 1, \ldots, 2s+4$, such that $t^* \in (\sigma_j, \tau_j)$ and $R^j$ is empty at times $\sigma_j$ and $\tau_j$ but contains pebbles throughout the interval $(\sigma_j, \tau_j)$ during which it is completely pebbled.

By Lemma 5.23 we know that less than $4s+4$ pyramids contain pebbles at time $\sigma_j$ and similarly at time $\tau_j$. Since all $c$ pyramids in $\Gamma(c,r)$ must have their sinks pebbled during $(\sigma_j, \tau_j)$ but we have $2 \cdot (4s+4) < c$ by the assumptions in Lemma 5.22, we conclude from Lemma 5.21 that for every interval $(\sigma_j, \tau_j)$ we can find some pyramid $\Pi^j$ that is completely pebbled during this interval. This, in turn, implies that there is some time $t_j^* \in (\sigma_j, \tau_j)$ when the pyramid $\Pi^j$ contains at least $\mathit{BW\text{-}Peb}^{\emptyset}(\Pi^j) = r+1$ pebbles. (We note that many $t_j^*$ can be equal and even refer to the same pyramid which has just happened to receive a lot of different labels, but this is not a problem as we shall see.)

As in the proof of Lemma 5.23, we now sort the $t_j^*$, $j = 1, \ldots, 2s+4$, in increasing order and consider the two possible cases. If at least half of all $t_j^*$ satisfy $t_j^* \leq t^*$, we look at the largest $t_j^* \leq t^*$. At this time there are at least $r+1$ pebbles on $\Pi^j$ and at least $\frac{2s+4}{2} = s+2$ pebbles on different sections, which means that $\mathit{space}(\mathbb{P}_{t_j^*}) \geq rs+3$ exceeds the space restrictions. If, on the other hand, at least half of all $t_j^*$ satisfy $t_j^* > t^*$, then for the smallest $t_j^*$ larger than $t^*$ we can again conclude that $\mathit{space}(\mathbb{P}_{t_j^*}) \geq r+s+3$, which is a contradiction. The lemma follows. □

Putting together everything that has been proven so far in this section, we are able to establish the pebbling trade-off result.

*Proof of Lemma 5.22.* Suppose that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling on $\Gamma(c, r)$ pebbling all sinks and that $\max\{\textit{space}(\mathbb{P}_\sigma), \textit{space}(\mathbb{P}_\tau)\} < s$ and $\textit{space}(\mathcal{P}) < (r + 2) + s$ for $0 < s \leq c/8 - 1$. Let us define

$$T(c, r, s) = \left(\frac{c - 2s}{4s + 4}\right)^r \cdot r! \quad . \tag{34}$$

We show that $\textit{time}(\mathcal{P}) \geq T(c, r, s)$ by induction over $r$.

For $r = 1$, the assumptions in the lemma imply that more than $c - 2s$ sinks are empty at times $\sigma$ and $\tau$. These sinks must be pebbled, which trivially requires strictly more than $c - 2s > \left(\frac{c-2s}{4s+4}\right) = T(c, 1, s)$ time steps.

Assume that the lemma holds for $\Gamma(c, r - 1)$ and consider any pebbling of $\Gamma(c, r)$. Less than $2s$ spines contain pebbles at time $\sigma$ or time $\tau$. All the other strictly more than $c - 2s$ spines are empty at times $\sigma$ and $\tau$ but must be completely pebbled during $[\sigma, \tau]$ by Lemma 5.10.

Consider the first time $\sigma'$ when any spine gets a pebble for the first time. Let us denote this spine by $Q'$. By Lemma 5.20 we know that $Q'$ contains pebbles during a contiguous time interval until it is completely pebbled and emptied at, say, time $\tau'$. During this whole interval $[\sigma', \tau']$ less than $4s + 4$ sections contain pebbles at any one given time, so in particular less then $4s+4$ spines contain pebbles. Moreover, Lemma 5.20 says that every spine containing pebbles will remain pebbled until completed. What this means is that if we order the spines with respect to the time when they first receive a pebble in groups of size $4s + 4$, no spine in the second group can be pebbled until the at least one spine in the first group has been completed.

We remark that this divides the spines that are empty at the beginning and end of $\mathcal{P}$ into strictly more than

$$\frac{c - 2s}{4s + 4} \tag{35}$$

groups. Furthermore, we claim that completely pebbling just one empty spine requires at least

$$r \cdot T(c, r - 1, s) \tag{36}$$

time steps. Given this claim we are done, since combining (35) and (36) we can deduce that the total pebbling time is lower-bounded by

$$\frac{c - 2s}{4s + 4} r \cdot T(c, r - 1, s) = T(c, r, s) \tag{37}$$

since at least one spine from each group is pebbled in a time interval totally disjoint from the time intervals for all spines in the next group.

It remains to establish the claim. To this end, fix any spine $Q^*$ empty at times $\sigma^*$ and $\tau^*$ but completely pebbled in $[\sigma^*, \tau^*]$. Consider the first time $\tau_1 \in [\sigma^*, \tau^*]$ when any section in $Q^*$, let us denote it by $R_1$, has been completely pebbled (i.e., , all vertices has been touched by pebbles but are now empty again). During $[\sigma^*, \tau_1]$ all pyramid sinks $z_1, \ldots, z_c$ are pebbled (Lemma 5.10), and since less than $2 \cdot (4s+4) < c$ pyramids contain pebbles at times $\sigma^*$ or $\tau_1$ (Lemma 5.23), at least one pyramid is pebbled completely (Lemma 5.21), which requires $r+1$ pebbles. Moreover, there is at least one pebble on $R_1$ during this whole interval. Hence, there is a time $\sigma_1 \in [\sigma^*, \tau_1]$ when there are strictly less than $(r + 2) + s - (r + 1) - 1 = s$ pebbles on $\Gamma(c, r - 1)$. Also, at this time $\sigma_1$ less than $4s + 4$ sections contain pebbles (Lemma 5.24), and in particular this means that there are pebbles on less than $4s + 3$ other section of our spine $Q^*$. This puts an upper bound on the number of sections of $Q^*$ pebbled this far, since every section is completely pebbled during a contiguous time interval before being emptied again, and we chose to focus on the first section $R_1$ in $Q^*$ that was finished.

Look now at the first section $R_2$ in $Q^*$ *other than the less than* $4s + 4$ *sections containing pebbles at time* $\sigma_1$ that is completely pebbled, and let the time when $R_2$ is finished be denoted $\tau_2$ (clearly, $\tau_2 > \tau_1$).

During $[\sigma_1, \tau_2]$ all sinks of $\Gamma(c, r-1)$ must have been pebbled, and at time $\tau_2 - 1$ less than $4s + 3$ other section in $Q^*$ contain pebbles.

Wrapping up, consider the first new section $R_3$ in our spine $Q^*$ to be completely pebbled among those that has not yet been touched at time $\tau_2 - 1$. Suppose that $R_3$ is finished at time $\tau_3$. Then during $[\tau_2, \tau_3]$ some pyramid is completely pebbled, and thus there must exist a time $\sigma_3 \in (\tau_2, \tau_3)$ when there are at least $r + 1$ pebbles on this pyramid and at least one pebble on the spine $Q^*$, leaving less than $s$ pebbles for $\Gamma(c, r-1)$. But this means that we can apply the induction hypothesis on the interval $[\sigma_1, \sigma_3]$ and deduce that $\sigma_3 - \sigma_1 \geq T(c, r-1, s)$. Note also that at time $\sigma_3$ less than $8s + 8 < c$ sections in $Q^*$ have been finished.

Continuing in this way, for every group of $8s + 8 < c$ finished sections in $Q^*$ we get one pebbling of $\Gamma(c, r-1)$ in space less than $\textit{BW-Peb}^{\emptyset}\bigl(\Gamma(c, r-1)\bigr) + s$ and with less than $s$ pebbles in the start and end configurations, which allows us to apply the induction hypothesis a total number of at least $\frac{cr}{8s+8} > r$ times. (Just to argue that we get the constants right, note that $8s + 8 < c$ implies that after the final pebbling of the sinks of $\Gamma(c, r-1)$ has been done, there is still some empty section left in $Q^*$. When this final section is taken care of, we will again get at least $r + 1$ pebbles on some pyramid while at least one pebble resides on $Q^*$, so we get the space on $\Gamma(c, r-1)$ down below $s$ as is needed for the induction hypothesis.)

This proves our claim that pebbling one spine takes time at least $r \cdot T(c, r-1, s)$. The lemma follows. ☐

As we already noted, this completes the proof of Theorem 5.19, since this theorem follows immediately from Lemma 5.22 for the special case when $\mathbb{P}_\sigma = \mathbb{P}_\tau = (\emptyset, \emptyset)$.

## 5.3 Recapitulation of Some Known Pebbling Trade-off Results

All the material in Section 5.3 is from [LT82]. The statements of the results below differ slightly in the constants in that paper, though, since there are some (minor) technical differences in the definitions as compared to the present paper.

### 5.3.1 Pebbling Trade-offs for Constant Space

Even for graphs pebblable in minimal constant space, there are nontrivial time-space trade-offs. More precisely, Lengauer and Tarjan [LT82] prove the following quadratic trade-offs for constant pebbling space.

**Theorem 5.25 ([LT82]).** *There are explicitly constructible DAGs $G_n$ of size $\Theta(n)$ with a single sink and maximal indegree 2 having the following pebbling properties:*

- *The black pebbling price of $G_n$ is $\mathsf{Peb}(G_n) = 3$.*
- *Any black pebbling strategy $\mathcal{P}_n$ for $G_n$ that optimizes time given space constraints*[6] *$\mathrm{O}(n)$ exhibits a trade-off $\mathsf{time}(\mathcal{P}_n) = \Theta\bigl(n^2/\mathsf{space}(\mathcal{P}_n)\bigr)$.*
- *Any black-white pebbling strategy $\mathcal{P}_n$ for $G_n$ that optimizes time given space constraints $\mathrm{O}\bigl(\sqrt{n}\bigr)$ exhibits a trade-off $\mathsf{time}(\mathcal{P}_n) = \Theta\bigl((n/\mathsf{space}(\mathcal{P}_n))^2\bigr)$.*

We will present (most of) the proof of Theorem 5.25, since we have to use this theorem in a "non-black-box" way to derive the results that we need. The trade-offs in the theorem are obtained for graphs built from permutations in the following way.

[6] The reason for including the upper bounds on space in the statement of the theorem is that no matter how much space is available, it is of course never possible to do better than linear time. Thus the trade-offs cannot hold when length dips below linear.

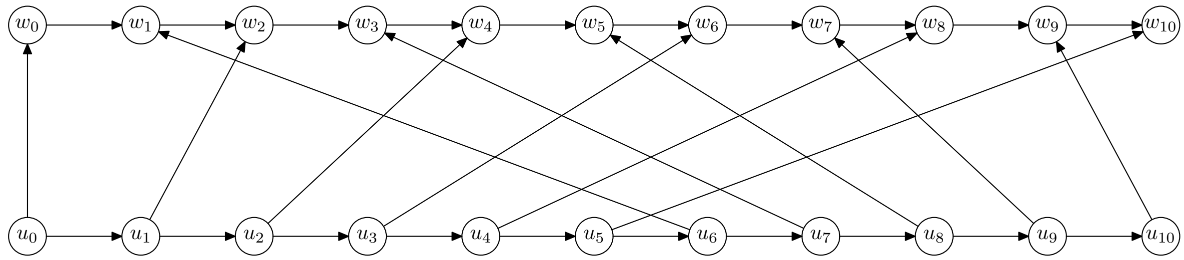


**Figure 5:** Permutation graph over $11$ vertices defined by permutation sending $x$ to $2x \bmod 11$.

**Definition 5.26 (Permutation graph ([LT82])).** Let $\pi$ denote some permutation of $\{0, 1, \ldots, n-1\}$. The *permutation graph* $\Delta(n, \pi)$ over $n$ elements with respect to $\pi$ is defined as follows. $\Delta(n, \pi)$ has $2n$ vertices divided into a *lower row* with vertices $u_0, u_1, \ldots, u_{n-1}$ and an *upper row* with vertices $w_0, w_1, \ldots, w_{n-1}$. For all $i = 0, 1, \ldots, n-2$, there are directed edges $(u_i, u_{i+1})$ and $(w_i, w_{i+1})$, and for all $i = 0, 1, \ldots, n-1$, there are edges $\left(u_i, w_{\pi(i)}\right)$ from the lower row to the upper row.

Thus, the only source vertex in $\Delta(n, \pi)$ is $u_0$ and the only sink vertex is $w_{n-1}$. All vertices in the lower row except the leftmost one have indegree 1 and all vertices in the upper row except the leftmost one have indegree 2. Figure 5 shows an example of a permutation graph.

Any DAG of fan-in 2 must have pebbling price at least 3. It is not too hard to see that permutation graphs $\Delta(n, \pi)$ have pebbling strategies in this minimal space: keeping one pebble on vertex $w_i$ of the upper row, move two pebbles consecutively on the lower row until $u_{\pi^{-1}(i+1)}$ is reached, and then pebble $w_{i+1}$. This strategy is not too efficient timewise, however. It will take time $\Omega(n^2)$ in the worst case (for instance, for the permutation sending $i$ to $n - i - 1$).

Generalizing the pebbling strategy just sketched, we get the following upper bound on the time-space trade-off for any permutation graph.

**Lemma 5.27 ([LT82]).** *Let $\Delta(n, \pi)$ be the permutation graph over $n$ elements for any permutation $\pi$. Then the black pebbling price of $\Delta(n, \pi)$ is $\mathsf{Peb}(\Delta(n, \pi)) = 3$, and for any space $s$, $3 \leq s \leq n$, there is a black pebbling strategy $\mathcal{P}$ for $\Delta(n, \pi)$ with $\mathsf{space}(\mathcal{P}) \leq s$ and $\mathsf{time}(\mathcal{P}) \leq \frac{2n^2}{s-2} + 2n$.*

Clearly, the space interval of interest is $3 \leq s \leq n$ since for $s > n$ there is the trivial pebbling that places pebbles on all vertices in the lower row and then sweeps a black pebble across the upper row.

To prove lower bounds for permutation graphs, Lengauer and Tarjan focus on permutations defined in terms of reversing the binary representation of the integers $\{0, 1, \ldots, n-1\}$ when $n$ is an even power of 2.

**Definition 5.28 (Bit reversal graph ([LT82])).** The *$m$-bit reversal* of $i$, $0 \leq i \leq 2^m - 1$, is the integer $\mathrm{rev}_m(i)$ obtained by writing the $m$-bit binary representation of $i$ in reverse order. The *bit reversal graph* $\Delta(2^m, \mathrm{rev}_m)$ is the permutation graph over $n = 2^m$ with respect to $\mathrm{rev}_m$.

For instance, we have $\mathrm{rev}_3(1) = 4$, $\mathrm{rev}_3(2) = 2$, and $\mathrm{rev}_3(3) = 6$. We will denote the bit reversal graph by $\Delta(n, \mathrm{rev})$ for simplicity, implicitly assuming that $n = 2^m$. An example of a bit reversal graph can be found in Figure 6.

For bit reversal graphs, the trade-off in Lemma 5.27 for black pebbling is asymptotically tight.

**Theorem 5.29 ([LT82]).** *Suppose that $\mathcal{P}$ is any complete black pebbling of the bit reversal graph $\Delta(n, \mathrm{rev})$ over $n = 2^m$ elements such that $\mathsf{space}(\mathcal{P}) = s$ for $s \geq 3$. Then $\mathsf{time}(\mathcal{P}) \geq \frac{n^2}{8s}$.*

Note, in particular, that if we want to black-pebble $\Delta(n, \mathrm{rev})$ in linear time, then linear space is needed. We again omit the proof in order to focus instead on the more challenging black-white pebbling case. It

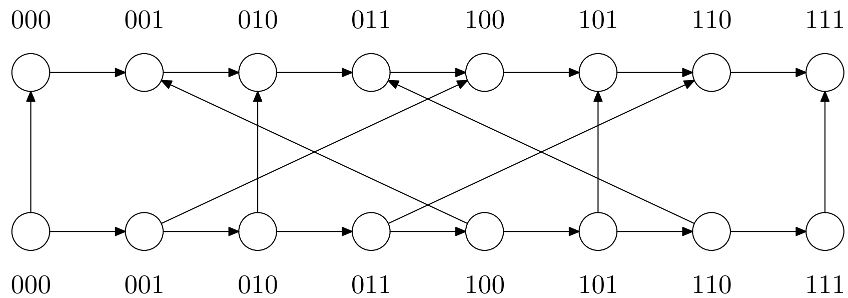


**Figure 6:** Bit reversal graph $\Delta(8, \mathrm{rev})$ on $8$ elements.

turns out that if we are also allowed to use white pebbles, the proof of Theorem 5.29 breaks down due to the fact that a central assumption in the proof is that any pebbling proceeds through the DAG in topological order. This does not hold for a black-white pebbling since white pebbles can be placed anywhere in the graph. Modifying the argument to take this possibility into account, we get the following lower bound.

**Theorem 5.30 ([LT82]).** *Let $\mathcal{P}$ be any complete black-white pebbling of $\Delta(n, \mathrm{rev})$ with $\mathsf{space}(\mathcal{P}) = s$ for $s \geq 3$. Then $\mathsf{time}(\mathcal{P}) \geq \frac{n^2}{18s^2} + 2n$.*

*Proof.* Suppose that $s < n/6$ since otherwise the statement is trivially true. Write $m = \log n$ and fix $r$ such that $3s \leq 2^r < 6s$. Divide the vertices in the upper row into $2^{m-r} > n/6s$ intervals

$$I_j = \{w_{j \cdot 2^r},\, w_{j \cdot 2^r + 1},\, \ldots,\, w_{(j+1) \cdot 2^r - 1}\} \tag{38}$$

of length $2^r$ for $0 \leq j < 2^{m-r}$. Let $\tau_0 = 0$ and $M_0 = \emptyset$, and inductively define $\tau_i$ to be the first time after $\tau_{i-1}$ when the first interval $I_j \notin M_{i-1}$ has been pebbled and unpebbled completely. At time $\tau_i$, a pebble is removed from $I_j$ and at most $s-1$ other intervals $I_{j'}$ contain pebbles. Let $M_i$ be the union of $M_{i-1}$ and the at most $s$ intervals just mentioned, including $I_j$. Repeat this procedure for $i = 1, 2, 3, \ldots$ until $M_i$ covers all intervals (which clearly must be the case at the end of the pebbling).

There are strictly more than $n/6s$ intervals, and at most $s$ new intervals are added to $M_i$ in each iteration. Hence, the above procedure is repeated at least $\lceil n/6s^2 \rceil$ times. We claim that in between $\tau_{i-1}$ and $\tau_i$, there are at least $n/6$ pebble placements made on the lower row. To prove this claim, note first that by construction $I_j$ is empty at time $\tau_{i-1}$, so all of $I_j$ is pebbled during $[\tau_{i-1}, \tau_i]$. Now look at the set of vertices

$$\mathrm{rev}_m^{-1}(I_j) = \{u_i \mid i = \mathrm{rev}_m^{-1}(j \cdot 2^r), \mathrm{rev}_m^{-1}(j \cdot 2^r + 1), \ldots, \mathrm{rev}_m^{-1}((j+1) \cdot 2^r - 1)\} \tag{39}$$

in the lower row. (Figure 7 illustrates $I_1 = \{w_4, w_5, w_6, w_7\}$ and $\mathrm{rev}_m^{-1}(I_1)$ for $r = 2$ in the bit reversal DAG over $16$ elements.) By the definition of bit reversal permutations, every $I_j$ divides the lower row into $2^r - 1$ intervals of length exactly $2^{m-r}$. To see this, note that $\mathrm{rev}_m^{-1}$ fixes the $n - r$ lower bits to the bit pattern $j \cdot 2^r$ reversed, while the $r$ upper bits run through all combinations of $0$ and $1$. Disregarding the leftmost and rightmost intervals, we get $2^r - 1$ intervals of length exactly $2^{m-r}$ in between the end intervals. At time $\tau_{i-1}$, at most $s-1$ of these intervals in the lower row contain pebbles, and at time $\tau_i$ at most $s-1$ other intervals contain pebbles. By Lemma 5.10, all the other at least $2^r - 2(s-1) > s$ intervals in the lower row must be completely pebbled and unpebbled during $[\tau_{i-1}, \tau_i]$. But this requires more than $s \cdot 2^{m-r} > s \cdot n/6s = n/6$ pebble placements.

Summing over all of the at least $\lceil n/6s^2 \rceil$ iterations, we get a total of more than $n/6 \cdot \lceil n/6s^2 \rceil \geq (n/6s)^2$ pebble placements on the lower row plus at least $n$ placements on the upper row, and multiplying by 2 to adjust for removals gives the bound stated in the theorem. ☐

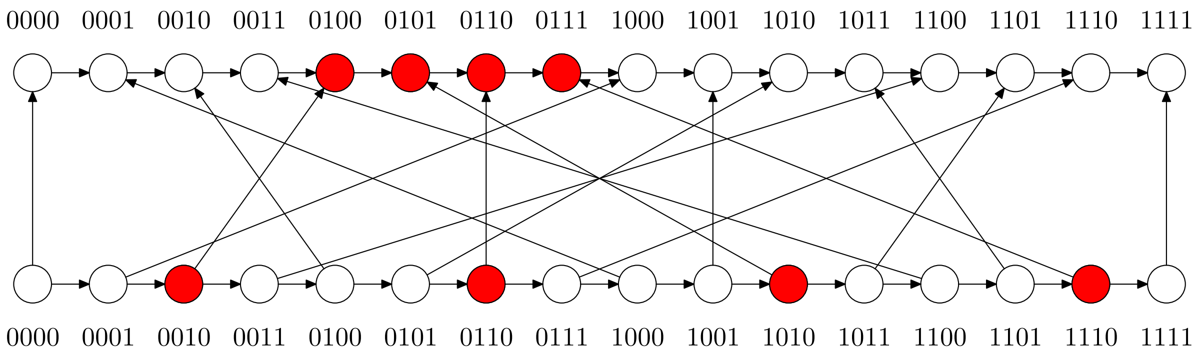


**Figure 7:** Upper-row vertices $w_{j\cdot 2^r}, w_{j\cdot 2^r+1}, \ldots, w_{(j+1)\cdot 2^r-1}$ split lower row into evenly sized intervals.

The reason for the discrepancy between Theorem 5.29 and Theorem 5.30 turns out to be that in fact, it *is* possible to do better using white pebbles in addition to the black ones. In particular, there is a linear-time black-white pebbling strategy for $\Delta(n, \mathrm{rev})$ using only order of $\sqrt{n}$ pebbles.

**Theorem 5.31 ([LT82]).** *For any space $s \geq 3$ there is a complete black-white pebbling $\mathcal{P}$ of $\Delta(n, \mathrm{rev})$ with space$(\mathcal{P}) \leq s$ and time$(\mathcal{P}) \leq 144\frac{n^2}{s^2} + 12n$.*

Since we will need to use the construction in Theorem 5.31 when devising resolution refutations of the corresponding pebbling formulas in Section 6.1, we present a detailed proof. The main work is in establishing the next lemma. We show the lemma first and then explain how it implies Theorem 5.31.

**Lemma 5.32 ([LT82]).** *For all $s$, $3 \leq s \leq 3\sqrt{n}$, there is a complete pebbling of $\Delta(n, \mathrm{rev})$ in space at most $s$ and time at most $144\frac{n^2}{s^2} + 2n$.*

*Proof of Lemma 5.32.* Write $m = \log n$ and let $r$ be the non-negative integer such that

$$3 \cdot 2^r \leq s < 3 \cdot 2^{r+1} \quad . \tag{40}$$

Divide the upper row of $\Delta(n, \mathrm{rev})$ into $2^r$ intervals

$$I_j = \left\{ w_{j\cdot 2^{m-r}+k} \,\middle|\, k = 0, 1, \ldots, 2^{m-r} - 1 \right\} \tag{41}$$

of size $2^{m-r}$ for $j = 0, \ldots, 2^r - 1$ and then subdivide each interval $I_j$ into $2^{m-2r}$ chunks by defining

$$C^i_j = \left\{ w_{j\cdot 2^{m-r}+i\cdot 2^r+k} \,\middle|\, k = 0, 1, \ldots, 2^r - 1 \right\} \tag{42}$$

for $i = 0, \ldots, 2^{m-2r} - 1$. Note that we must have $2^{m-2r} \geq 1$ for this definition to make sense, but this holds since $s \leq 3\sqrt{n}$ by assumption. Figure 8 exemplifies these definitions on the 32-element bit reversal DAG with $2^2$ intervals and 2 chunks per interval.

The pebbling strategy will proceed in $2^{m-2r}$ *phases* corresponding to the $2^{m-2r}$ chunks in each interval, and in $2^r$ *stages* within each phase corresponding to the different intervals. All the phases in the pebbling are completely analogous except for some minor tweaks in the first and final phases, which we refer to as the 0th and $(2^{m-2r} - 1)$st phases, respectively. To help the reader parse the notation, we note that in what follows superscripts $i$ will correspond to phases/chunks and subscripts $j$ to stages/intervals. We reserve $2^r$ pebbles for the lower row, $2^r$ pebbles for the "current chunks" in the upper row, and $2^r - 1$ pebbles for the rightmost vertices in $I_0, I_1, \ldots, I_{2^r-2}$. By the leftmost inequality in (40), this leaves one auxiliary pebble to help with advancing the other pebbles.

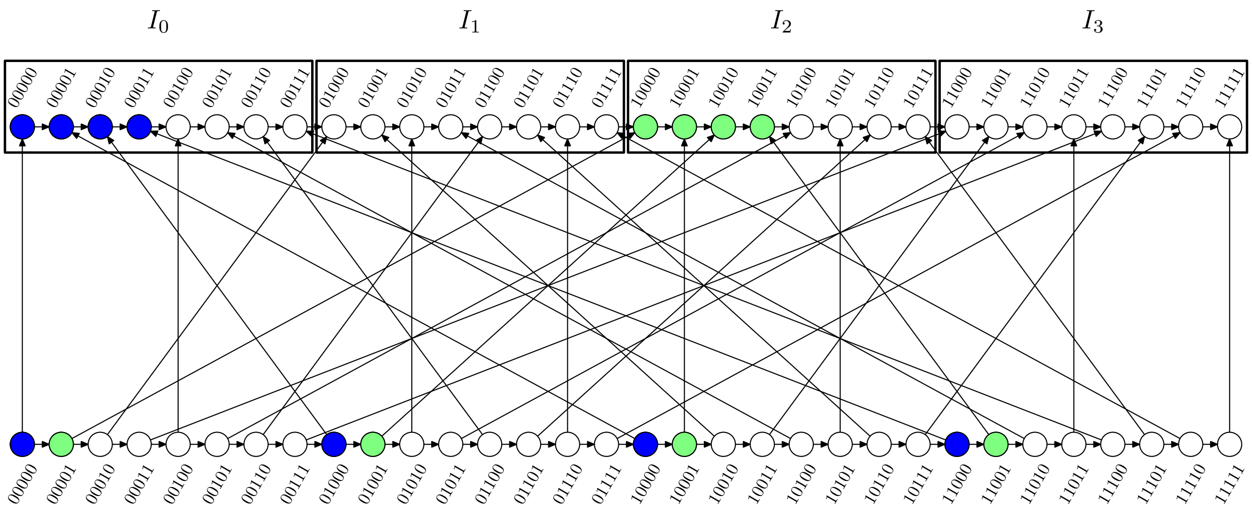


**Figure 8:** Intervals $I_j$ for $r = 2$ in $\Delta(32, \mathrm{rev})$ and $0$th chunks in $I_0$ and $I_{\mathrm{rev}_r(1)} = I_2$ with inverse images.

Start the $0$th stage in the $0$th phase by doing a black-only pebbling of the lower row, leaving pebbles on the $2^r$ vertices in

$$U_0^0 = \{u_{\mathrm{rev}_m(k)} \mid k = 0, 1, \ldots, 2^r - 1\} \tag{43}$$

and then, using the support of these pebbles, sweep a black pebble past the $0$th chunk $w_0, w_1, \ldots, w_{2^r-2}$ of $I_0$, leaving it on the rightmost vertex $w_{2^r-1}$. This concludes the $0$th stage.

In the next stage, move all black pebbles in $U_0^0$ on the lower row exactly one step to the right to the vertices $u_k$ for $k = 1, \mathrm{rev}_m(1) + 1, \mathrm{rev}_m(2) + 1, \ldots, \mathrm{rev}_m(2^r - 1) + 1$. Using the fact that we can write $1 = \mathrm{rev}_m(\mathrm{rev}_r(1) \cdot 2^{m-r})$ by shifting $1$ first $r$ bits to the left, then $m - r$ bits more and finally all the way back again, we see that the set of lower-row vertices now covered by black pebbles is

$$U_1^0 = \left\{u_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r}+k)} \,\middle|\, k = 0, 1, \ldots, 2^r - 1\right\} \ , \tag{44}$$

which by (42) is the set of all predecessors in the lower row of the $0$th chunk $C^0_{\mathrm{rev}_r(1)}$ of the interval $I_{\mathrm{rev}_r(1)}$ (see Figure 8 for a concrete example of this). If we place a white pebble on the rightmost vertex of the interval $I_{\mathrm{rev}_r(1)-1}$, this white pebble plus the lower-row black pebbles on $U_1^0$ allow us to advance a black pebble along all the vertices of the $0$th chunk of $I_{\mathrm{rev}_r(1)}$, leaving it on the rightmost vertex. This concludes stage $1$ of phase $0$.

Continuing in this way, in the $j$th stage of phase $0$ we can move the lower-row pebbles from $U^0_{j-1}$ to $U^0_j$ where this notation is generalized to mean

$$U_j^0 = \left\{u_{\mathrm{rev}_m(\mathrm{rev}_r(j)\cdot 2^{m-r}+k)} \,\middle|\, k = 0, 1, \ldots, 2^r - 1\right\} \tag{45}$$

for all $j \leq 2^r - 1$, and then place black pebbles on the rightmost vertex in every chunk $C^0_{\mathrm{rev}_r(j)}$ with the help of a white pebble on the rightmost vertex in $I_{\mathrm{rev}_r(j)-1}$. At the end of the final stage of phase $0$, we thus have black pebbles on the rightmost vertices of all $0$th chunks and white pebbles on the rightmost vertices of $I_0, I_1, \ldots, I_{2^r-2}$. Later phases will move the black pebbles to the right, chunk by chunk, while leaving the white pebbles in place. We observe that during phase $0$, we made at most $n$ pebble placements on the lower row to get the pebbles into “starting position” $U_0^0$, and then exactly $2^r$ placements more on the lower row in each of the other $2^r - 1$ stages.

Inductively, suppose at the beginning of phase $i$ that there are black pebbles on the rightmost vertices in all $(i - 1)$st chunks. Let us extend the lower-row vertex set notation above to full generality and define

$$U_j^i = \left\{u_{\mathrm{rev}_m(\mathrm{rev}_r(j)\cdot 2^{m-r}+i\cdot 2^r+k)} \,\middle|\, k = 0, 1, \ldots, 2^r - 1\right\} = \mathrm{rev}_m^{-1}\left(C^i_{\mathrm{rev}_r(j)}\right) \ , \tag{46}$$

where the second equality is easily verified from (42). In stage $0$ of phase $i$, we rearrange the lower-row black pebbles so that they cover the vertices in $U_0^i$. Since the $2^r$ black pebbles are already present somewhere on the lower row, this can be achieved with at most $n - 2^r$ pebble placements (the details can be found in the proof in [LT82] for our Lemma 5.27). This allows us to advance the pebble in $I_0$ on the upper row from the rightmost vertex in chunk $i-1$ to the rightmost vertex in chunk $i$. Moving the vertices in $U_0^i$ one step to the right in each following stage to $U_1^i, U_2^i$, et cetera, we can sweep black pebbles across the $i$th chunks of the other intervals $I_j$ in the order $I_{\mathrm{rev}_r(1)}, I_{\mathrm{rev}_r(2)}, \ldots, I_{\mathrm{rev}_r(2^r-2)}, I_{\mathrm{rev}_r(2^r-1)} = I_{2^r-1}$. All in all, we make at most $(n - 2^r) + (2^r - 1) \cdot 2^r$ pebble placements on the lower row during phase $i$ for $i \geq 1$.

In the final $(2^{m-2r} - 1)$st phase, we already have white pebbles on the rightmost vertex of the chunk in every interval except the rightmost one $I_{2^r-1}$. Therefore, in every stage except the final one, instead of placing a black pebble on the rightmost vertex in the chunk we use the black pebbles on the two predecessors of this vertex to remove the white pebble. In the very final stage, we place a black pebble on $w_{n-1}$. Removing all other pebbles from the DAG, which are all black, we have obtained a complete pebbling of $\Delta(n, \mathrm{rev})$.

The space of this pebbling is $3 \cdot 2^r \leq s$ by construction. As to pebble placements, it is easy to verify that each vertex in the upper row is pebbled exactly once. The number of pebble placements in the lower row is at most $n + (2^r - 1) \cdot 2^r$ during phase $0$ and at most $(n - 2^r) + (2^r - 1) \cdot 2^r$ for each of the other $2^{m-2r} - 1$ phases, and summing up we get a total of at most

$$
\begin{aligned}
2^{m-2r}\big((n - 2^r) + (2^r - 1) \cdot 2^r\big) + 2^r + 2n &< 2^{m-2r}\big(n + 2^{2r}\big) + 2n \\
&\leq 72\frac{n^2}{s^2} + 2n
\end{aligned}
\tag{47}
$$

placements, where we used that $2^{m-2r} \geq 1$, $2^r \leq s/3 < 2^{r+1}$, and $s \leq 3\sqrt{n}$. Multiplying by $2$ to take the pebble removals into account gives the time bound stated in the lemma. ☐

*Proof of Theorem 5.31.* For $s \leq 3\sqrt{n}$ the statement was proven in Lemma 5.32 (and note that for $s < 70$, the black-only pebbling in Lemma 5.27 gives a better time bound). To get the statement for $s > 3\sqrt{n}$, use the same pebbling strategy as in the proof of Lemma 5.32 but choose $r$ so that $\sqrt{n}/2 < 2^r \leq \sqrt{n}$. Then the number of chunks $2^{m-2r}$ is at most 2, and the time bound derived from (47) reduces to $12n$. ☐

On a high level, the reason that black-white pebblings can do much better than black-only pebblings on bit reversal DAGs is that these graphs have such a regular structure. Lengauer and Tarjan raise the question whether there are other permutations for which the lower bound in Theorem 5.29 holds also for black-white pebbling, and conjecture that the answer is yes. To the best of our knowledge, this problem is still open. We do not know of any candidate permutations for establishing the conjecture, but one could ask whether anything informative could be said about what holds for, for instance, a random permutation in this respect. If the conjecture turns out to be true for a random permutation (with high probability, say), then such a result, although non-constructive, would be interesting.

### 5.3.2 DAGs Yielding Robust Pebbling Trade-offs

To get robust pebbling trade-offs, i.e., trade-offs that hold over a large space interval, we use a DAG family studied in [LT82, Section 4].

**Definition 5.33 (Stack of superconcentrators ([LT82])).** Let $SC_m$ denote any (explicitly constructible) linear-size $m$-superconcentrator with bounded indegree and depth $\mathrm{O}(\log m)$. Then $\Phi(m, r)$ denotes the graph constructed by stacking $r$ copies $SC_m^1, \ldots, SC_m^r$ of $SC_m$ on top of one another, with the sinks $z_1^j, z_2^j, \ldots, z_m^j$ of $SC_m^j$ connected to the sources $s_1^{j+1}, s_2^{j+1}, \ldots, s_m^{j+1}$ of $SC_m^{j+1}$ by edges $\big(z_i^j, s_i^{j+1}\big)$ for all $i = 1, \ldots, m$ and all $j = 1, \ldots, r-1$.

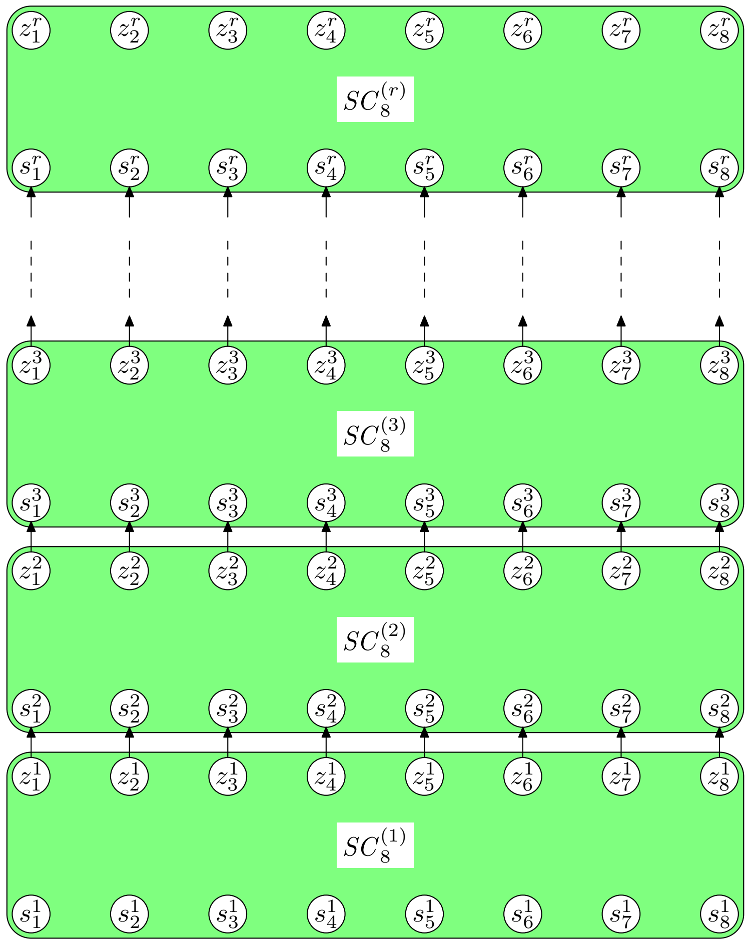


**Figure 9:** Schematic illustration of stack of superconcentrators $\Phi(8, r)$.

Clearly, $\Phi(m, r)$ has size $\Theta(rm)$. Figure 9 gives a schematic illustration of the construction.

Lengauer and Tarjan establish fairly detailed trade-off results for stacks of superconcentrators using different explicit and non-explicit constructions for the superconcentrator building blocks. All of these results can be translated into corresponding trade-off results in resolution. For simplicity and conciseness, however, we only state a special case of their results, and provide a brief proof sketch for the interested reader.

**Theorem 5.34 ([LT82]).** *For $\Phi(m, r)$ a stack of (explicitly constructible) linear-size $m$-superconcentrators with bounded indegree and depth $\mathrm{O}(\log m)$, the following holds:*

- $\mathit{Peb}\big(\Phi(m, r)\big) = \mathrm{O}(r \log m)$.
- *There is a linear-time black pebbling strategy $\mathcal{P}$ for $\Phi(m, r)$ with space$(\mathcal{P}) = \mathrm{O}(m)$.*
- *If $\mathcal{P}$ is a black-white pebbling strategy for $\Phi(m, r)$ in space $s \leq m/20$, then time$(\mathcal{P}) \geq m \cdot \left(\frac{rm}{64s}\right)^r$.*

*Proof sketch.* The upper bound on black pebbling price follows from Observation 5.9, since the depth of $\Phi(m, r)$ is $\mathrm{O}(r \log m)$. The linear-time black pebbling strategy is obtained by applying the trivial pebbling strategy in Observation 5.7 consecutively to each superconcentrator, keeping pebbles on the sinks of $SC_m^j$ while pebbling $SC_m^{j+1}$.

The reason that the final trade-off result holds is, very loosely put, that the lower bounds in Lemma 5.14 and Corollary 5.15 propagate through the stack of superconcentrators and get multiplied at each level. If the pebbling strategy is restricted to keeping $s/r$ pebbles on each copy $SC^j_m$ of the superconcentrator, this is not hard to prove directly from Lemma 5.14. Establishing that this intuition holds also in the general case, when pebbles may be unevenly distributed over the superconcentrator copies, is much more technically challenging, however. ☐

### 5.3.3 Exponential Pebbling Trade-offs

To get exponential trade-offs, i.e., trade-offs with lower bounds on the length on the form $2^{n^\epsilon}$ for some constant $\epsilon > 0$, the graphs in Section 5.3.2 are not sufficient. Instead, we need to appeal to stronger results from [LT82, Section 5].

**Theorem 5.35 ([LT82]).** *For every $\ell \in \mathbb{N}^+$ there exist constants $c, c' > 1$ such that the following holds for all sufficiently large $n$. Let $G$ be a DAG with $n$ vertices and maximal indegree $\ell$. Then for any space constraint $s$ satisfying $cn/\log n \leq s \leq n$, there is a black pebbling strategy $\mathcal{P}$ for $G$ with $\mathsf{space}(\mathcal{P}) \leq s$ and $\mathsf{time}(\mathcal{P}) \leq s \cdot 2^{2^{c'n/s}}$.*

By stacking superconcentrators of *defferent sizes* on top of one another, Lengauer and Tarjan are able to prove a matching lower bound. We refer to [LT82, Section 5] for the details of the construction.

**Theorem 5.36 ([LT82]).** *There exists a constant $\epsilon > 0$ such that the following holds for all sufficiently large integers $n, s$ satisfying $cn/\log n \leq s \leq n$: There exists a DAG $G$ with maximal indegree $2$ and number of vertices at most $n$ such that any black-white pebbling strategy $\mathcal{P}$ for $G$ with $\mathsf{space}(\mathcal{P}) \leq s$ must have $\mathsf{time}(\mathcal{P}) \geq s \cdot 2^{2^{\epsilon n/s}}$.*

Note that the graph $G$ in Theorem 5.36 depends on the pebbling space parameter $s$. Lengauer and Tarjan conjecture that no single graph gives an exponential time-space tradeoff for the whole range of $s \in [n/\log n, n]$, but to the best of our knowledge this problem is still open.

# 6 Time-Space Trade-offs for Resolution

We have finally reached the point where we can state and prove our time-space trade-off results for resolution. Given all the work done so far, the proofs are mostly simple variations of the following pattern: pick some graph family in Section 5, make the appropriate choices of parameters, consider the corresponding pebbling contradiction CNF formulas, do $f$-substitution for some non-authoritarian function $f$, and apply Theorem 4.4 (which we obtained with the help of Theorem 3.5).

Note that all the pebbling trade-off results are for explicit formulas (since they are pebbling formulas over explicitly constructible graphs). We also repeat one final time that all trade-offs hold for variable space and clause space simultaneously, since the upper bounds are for variable space and the lower bounds for clause space.

## 6.1 Trade-offs for Constant Space

Our first result is that time-space trade-offs in resolution can occur even for formulas refuted in (very small) constant space. What is more, there are such formulas for which we can prove not only a trade-off threshold, but even specify the whole trade-off curve.

**Theorem 6.1.** *There are explicitly constructible families $\{F_n\}_{n=1}^{\infty}$ of minimally unsatisfiable $k$-CNF formulas of size $\Theta(n)$ such that:*

1. *Every formula $F_n$ is refutable in resolution in length $L(F_n \vdash 0) = \mathrm{O}(n)$ and also in variable space $\mathit{VarSp}(F_n \vdash 0) = \mathrm{O}(1)$ (but not simultaneously).*

2. *For any $s > 0$ there is a refutation $\pi_n : F_n \vdash 0$ in simultaneous variable space $\mathit{VarSp}(\pi_n) = \mathrm{O}(s)$ and length $L(\pi) = \mathrm{O}\big((n/s)^2 + n\big)$.*

3. *Any resolution refutation $\pi_n : F_n \vdash 0$ in clause space $\mathit{Sp}(\pi_n) = s$ for $s \geq \mathit{Sp}(F_n \vdash 0)$ must have length $L(\pi) = \Omega\big((n/s)^2 + n\big)$.*

*The constants hidden in the asymptotic notation are independent of $n$ and $s$.*

*Proof.* Fix any non-authoritarian function $f$ of arity $d$ and consider the pebbling formulas $\mathit{Peb}_{\Delta(m,\mathrm{rev})}[f]$ defined over bit reversal DAGs $\Delta(m, \mathrm{rev})$ in Definition 5.28 for $m = \log n$.

Appealing to Theorem 4.4 will get us a long way but not quite to our final destination. More precisely, the upper bounds on length and space follow from Lemma 5.27 in this way, and the lower bound in the trade-off follows from Theorem 5.30. We cannot get the upper bound in the same manner, though, since Theorem 5.29 tells us that there *cannot* exist black pebblings with parameters matching the lower bounds for black-white pebblings. Obviously, if we could obtain a resolution refutation mimicking the black-white pebbling strategy in Theorem 5.31, we would get a tight trade-off result, but there is no known way of transforming black-white pebblings in general into resolution refutations with the same time and space parameters (and indeed we believe that this is not possible). However, in this particular case it turns out that we can construct a resolution refutation that simulates the black-white pebbling strategy in Theorem 5.31 in a space-preserving way. The rest of this proof is devoted to showing how this can be done.

Let us adopt all notation in the proofs of Theorem 5.31 and Lemma 5.32. In particular, we choose $r$ in the same way and then divide the upper row of $\Delta(n, \mathrm{rev})$ into $2^r$ intervals $I_j = \big\{w_{j \cdot 2^{m-r}+k} \,\big|\, k = 0, 1, \ldots, 2^{m-r} - 1\big\}$ of size $2^{m-r}$ for $j = 0, \ldots, 2^r - 1$ as in (41) and further subdivide each interval into $2^{m-2r}$ chunks $C^i_j = \big\{w_{j \cdot 2^{m-r}+i \cdot 2^r+k} \,\big|\, k = 0, 1, \ldots, 2^r - 1\big\}$ for $i = 0, \ldots, 2^{m-2r} - 1$ as in (42). Recall that this notation was illustrated in Figure 8 on page 43. We also remind the reader of the definition in (46) of the vertex sets $U^i_j = \big\{u_{\mathrm{rev}_m(\mathrm{rev}_r(j) \cdot 2^{m-r}+i \cdot 2^r+k)} \,\big|\, k = 0, 1, \ldots, 2^r - 1\big\}$ in the lower row which can be seen to be inverse images of the chunks $C^i_{\mathrm{rev}_r(j)}$ in the upper row.

Our resolution refutation will following the pebbling strategy described in the proof of Lemma 5.32 closely and proceed in $2^{m-2r}$ phases (numbered $0, 1, \ldots, 2^{m-2r} - 1$) corresponding to the $2^{m-2r}$ chunks in each interval, and in $2^r$ stages (numbered $0, 1, \ldots, 2^r - 1$) within each phase corresponding to the different intervals. All the phases in the refutation follow the same pattern except for some minor differences in the first and final phases, which we refer to as the 0th and $(2^{m-2r} - 1)$st phases, respectively. We will reserve $d \cdot 2^{3d} \cdot 2^r$ variable space for the lower row, $d \cdot 2^{3d} \cdot 2^r$ variable space for the "current chunks" in the upper row, and additional variable space $d \cdot 2^{3d}$ for each of the $2^r$ intervals $I_j$, which by the way we have chosen $r$ sums to a total of $\mathrm{O}\big(d \cdot 2^{3d} \cdot s\big) = \mathrm{O}(s)$ variable space when $d$ is fixed.

Using the notation for substitution formulas in Definition 3.2, a black pebble on a vertex $v$ in our translation of the black-white pebbling to resolution will be interpreted as having all clauses in $v[f]$ in memory, and a white pebble on $v$ will be interpreted as all the clauses $\overline{v}[f]$. We will use the notation

$$\mathbb{D}^i_j = \big\{v[f] \,\big|\, v \in U^i_j\big\} = \big\{\big(u_{\mathrm{rev}_m(\mathrm{rev}_r(j) \cdot 2^{m-r}+i \cdot 2^r+k)}\big)[f] \,\big|\, k = 0, 1, \ldots, 2^r - 1\big\} \tag{48}$$

for the set of clauses intuitively corresponding to black pebbles on all vertices in $U^i_j$. Also recall that for two clause sets $\mathbb{C}$ and $\mathbb{D}$, the notation $\mathbb{C} \vee \mathbb{D}$ is shorthand for $\{C \vee D \mid C \in \mathbb{C},\, D \in \mathbb{D}\}$.

We start stage 0 in phase 0 by deriving all clauses in $\mathbb{D}^0_0 = \{u_{\mathrm{rev}_m(k)}[f] \mid k = 0, 1, \ldots, 2^r - 1\}$ by imitating a black-only pebbling of the lower row leaving pebbles on the vertices in $U^0_0$. This can be done essentially in variable space $d \cdot 2^{2d} \cdot 2^r$. We refer to the the translation of black pebblings to resolution refutations in the proof of Theorem 4.1 for the details. Downloading all axioms in $(\overline{u}_0 \vee w_0)[f]$ and using

$u_0[f] \subseteq \mathbb{D}^0_0$, we can derive $w_0[f]$. Then, for each $k = 1, 2, \ldots, 2^r - 1$ in turn, we download all axioms in $(\overline{u}_{\mathrm{rev}_m(k)} \lor \overline{w}_{k-1} \lor w_k)[f] = \overline{u}_{\mathrm{rev}_m(k)}[f] \lor \overline{w}_{k-1}[f] \lor w_k[f]$ and, using the clauses $u_{\mathrm{rev}_m(k)}[f] \subseteq \mathbb{D}^0_0$ as well as the clauses $w_{k-1}[f]$ just derived, resolve over all variables in $u_{\mathrm{rev}_m(k)}[f]$ and $w_{k-1}[f]$ to get the clause set $w_k[f]$, after which all clauses in $w_{k-1}[f]$ are erased. In this way, we finally arrive at the clause set $w_{2^r-1}[f]$ which is the parallel of a black pebble on the rightmost vertex $w_{2^r-1}$ in the 0th chunk of $I_0$. This concludes the 0th stage of phase 0.

In the next stage of phase 0, we use the clauses in the set $\mathbb{D}^0_0$ as well as all the axiom clauses in $(\overline{u}_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r}+k)-1} \lor u_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r}+k)})[f]$, $k = 0, 1, \ldots, 2^r - 1$, to derive the clause set $\mathbb{D}^0_1$, after which all clauses in $\mathbb{D}^0_0$ are erased. This correponds to shifting all black pebbles on the vertices in $U^0_0$ one step to the right to $U^0_1 = \bigl\{u_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r}+k)} \bigm| k = 0, 1, \ldots, 2^r - 1\bigr\}$. (We remind the reader that this step is illustrated in Figure 8 on page 43.) When we are done with this, we download all axiom clauses in $(\overline{u}_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r})} \lor \overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \lor w_{\mathrm{rev}_r(1)\cdot 2^{m-r}})[f]$ and resolve with $u_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r})}[f] \subseteq \mathbb{D}^0_1$ to obtain $(\overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \lor w_{\mathrm{rev}_r(1)\cdot 2^{m-r}})[f]$. In pebbling terms, this corresponds to placing a white pebble on the rightmost vertex $w_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1}$ of the interval $I_{\mathrm{rev}_r(1)-1}$ and a black pebble on the leftmost vertex $w_{\mathrm{rev}_r(1)\cdot 2^{m-r}}$ of the interval $I_{\mathrm{rev}_r(1)}$. This black pebble placement is legal in view of the black pebble on $u_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r})}$, corresponding to the clause set $u_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r})}[f]$.

Pattern matching on what was done in stage 0, by induction over $k = 1, 2, \ldots, 2^r - 1$ we download all axioms in $(\overline{u}_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r}+k)} \lor \overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}+k-1} \lor w_{\mathrm{rev}_r(1)\cdot 2^{m-r}+k})[f]$ and then use the clauses $u_{\mathrm{rev}_m(\mathrm{rev}_r(1)\cdot 2^{m-r}+k)}[f] \subseteq \mathbb{D}^0_1$ as well as the clauses $(\overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \lor w_{\mathrm{rev}_r(1)\cdot 2^{m-r}+k-1})[f]$ derived by induction to infer the clause set $(\overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \lor w_{\mathrm{rev}_r(1)\cdot 2^{m-r}+k})[f]$. When this has been done, the clauses $(\overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \lor w_{\mathrm{rev}_r(1)\cdot 2^{m-r}+k-1})[f]$ are erased. This can be seen to resemble advancing a black pebble along all the vertices of the 0th chunk of $I_{\mathrm{rev}_r(1)}$, leaving it on the rightmost vertex of the chunk at the end of stage stage 1 of phase 0.

Continuing in this way, in the $j$th stage of phase 0 we use the clauses in $\mathbb{D}^0_{j-1}$ to derive the clause set $\mathbb{D}^0_j$ and then erase all of $\mathbb{D}^0_{j-1}$, which corresponds to moving the lower-row black pebbles from $U^0_{j-1}$ to $U^0_j = \bigl\{u_{\mathrm{rev}_m(\mathrm{rev}_r(j)\cdot 2^{m-r}+k)} \bigm| k = 0, 1, \ldots, 2^r - 1\bigr\}$. Then we mimic the placement of a black pebble on the rightmost vertex in the chunk $C^0_{\mathrm{rev}_r(j)}$ with the help of a white pebble on the rightmost vertex in $I_{\mathrm{rev}_r(j)-1}$ by downloading all all axioms in $(\overline{u}_{\mathrm{rev}_m(\mathrm{rev}_r(j)\cdot 2^{m-r})} \lor \overline{w}_{\mathrm{rev}_r(j)\cdot 2^{m-r}-1} \lor w_{\mathrm{rev}_r(j)\cdot 2^{m-r}})[f]$. Finally, we simulate the sweeping of a black pebble across all of $C^0_{\mathrm{rev}_r(j)}$ by performing derivation steps analogous to those in stages 0 and 1 described above to infer the clauses $(\overline{w}_{\mathrm{rev}_r(j)\cdot 2^{m-r}-1} \lor w_{\mathrm{rev}_r(j)\cdot 2^{m-r}+2^r-1})[f]$.

At the end of the final stage of phase 0, we thus have the clauses $w_{2^r-1}[f]$ as well as all clauses $(\overline{w}_{j\cdot 2^{m-r}-1} \lor w_{j\cdot 2^{m-r}+2^r-1})[f]$ for $j = 1, 2, \ldots, 2^r - 1$. This is our way of matching the black pebbles on the rightmost vertices of all 0th chunks and white pebbles on the rightmost vertices of all intervals except the last one at the end of phase 0 in the pebbling of Lemma 5.32.

Inductively, suppose at the beginning of the $i$th phase that the clause configuration contains the clauses $w_{i\cdot 2^r-1}[f]$ as well as all clauses $(\overline{w}_{j\cdot 2^{m-r}-1} \lor w_{j\cdot 2^{m-r}+i\cdot 2^r-1})[f]$ for $j = 1, 2, \ldots, 2^r - 1$. In terms of pebbles, this means that the white pebbles on the rightmost elements of all intervals except the last are still in place while the black pebble in each interval has moved along to the rightmost vertex of the $(i-1)$st chunk.

In stage 0 of phase $i$, we derive the clauses $\mathbb{D}^i_0$, corresponding to a rearrangement of the lower-row black pebbles so that they cover the vertices in $U^i_0$. Mimicking the subpebbling advancing the black pebble in $I_0$ on the upper row from the rightmost vertex in chunk $i-1$ to the rightmost vertex in chunk $i$, we use the clauses $w_{i\cdot 2^r-1}[f]$ and $\mathbb{D}^i_0$ to infer the clauses $w_{(i+1)\cdot 2^r-1}[f]$. In the following stages, the pebbling strategy moves the pebbles in $U^i_0$ one step to the right in each stage to $U^i_1, U^i_2$, et cetera, and sweeps black pebbles across the $i$th chunks of the other intervals $I_j$ in the order $I_{\mathrm{rev}_r(1)}, I_{\mathrm{rev}_r(2)}, \ldots, I_{\mathrm{rev}_r(2^r-2)}, I_{\mathrm{rev}_r(2^r-1)} = I_{2^r-1}$. Our resolution refutation under construction simulates this by deriving $\mathbb{D}^i_1, \mathbb{D}^i_2, \ldots, \mathbb{D}^i_{2^r-2}, \mathbb{D}^i_{2^r-1}$, and using each such clause set $\mathbb{D}^i_j$ to infer $(\overline{w}_{j\cdot 2^{m-r}-1} \lor w_{j\cdot 2^{m-r}+(i+1)\cdot 2^r-1})[f]$ from $(\overline{w}_{j\cdot 2^{m-r}-1} \lor w_{j\cdot 2^{m-r}+i\cdot 2^r-1})[f]$ in the order $j = \mathrm{rev}_r(1), \mathrm{rev}_r(2), \ldots, \mathrm{rev}_r(2^r-2), 2^r-1$.

Consider now the final $(2^{m-2r}-1)$st phase. In the pebbling strategy, we had to take care of a special case here since there are already white pebbles on the rightmost vertex of the chunk in every interval except the rightmost one $I_{2^r-1}$. Therefore, in every stage except the final one, instead of placing a black pebble on the rightmost vertex in the chunk, the pebbling strategy uses the black pebbles on the two predecessors of this vertex to remove the white pebble. We need to do something similar in spirit in our resolution refutation. Rather than getting lost in even more indices than we already have, let us describe somewhat informally how the final phase of the refutation proceeds.

At the beginning of the phase, the clause configuration contains the clauses $w_{2^{m-r}-2^r-1}[f]$ as well as all clauses $(\overline{w}_{j\cdot 2^{m-r}-1} \vee w_{(j+1)\cdot 2^{m-r}-2^r-1})[f]$ for $j = 1, 2, \ldots, 2^r-1$. At the end of stage $0$, we have derived the clause set $w_{2^{m-r}-1}[f]$. We resolve all clauses in this clause set with $(\overline{w}_{2^{m-r}-1} \vee w_{2\cdot 2^{m-r}-2^r-1})[f]$ to infer $w_{2\cdot 2^{m-r}-2^r-1}[f]$. Intuitively, this resembles the way the white pebble on $w_{2^{m-r}-1}$ is eliminated in the pebbling strategy.

In stage 1, we move on to the interval $I_{\mathrm{rev}_r(1)}$. At the beginning of the stage we have the clauses $(\overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \vee w_{(\mathrm{rev}_r(1)+1)\cdot 2^{m-r}-2^r-1})[f]$ in memory, and the stage ends with the derivation of the clauses $(\overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \vee w_{(\mathrm{rev}_r(1)+1)\cdot 2^{m-r}-1})[f]$. We can resolve these newly derived clauses with the clauses $(\overline{w}_{(\mathrm{rev}_r(1)+1)\cdot 2^{m-r}-1} \vee w_{(\mathrm{rev}_r(1)+2)\cdot 2^{m-r}-2^r-1})[f]$, available in memory by the induction hypothesis, to obtain $(\overline{w}_{\mathrm{rev}_r(1)\cdot 2^{m-r}-1} \vee w_{(\mathrm{rev}_r(1)+2)\cdot 2^{m-r}-2^r-1})[f]$. This is the intuitive parallel of removing the white pebble from $w_{(\mathrm{rev}_r(1)+1)\cdot 2^{m-r}-1}$.

Continuing in this way with the intervals $I_j$ in the order $j = \mathrm{rev}_r(2), \mathrm{rev}_r(3), \ldots, \mathrm{rev}_r(2^r-2), 2^r-1$, we finally obtain the clause set $w_{n-1}[f]$. Downloading all sink axioms $\overline{w}_{n-1}[f]$, we can infer the empty clause. The resolution refutation is thus complete.

It is straightforward, if tedious, to verify that the length and variable space of this resolution refutation are as claimed in Theorem 6.1. Again we refer to (the proof of) Theorem 4.1 for the details. ☐

## 6.2 Superpolynomial Trade-offs for any Non-constant Space

It is clear that we can never get superpolynomial trade-offs from DAGs pebblable in constant space, since such graphs must have constant-space pebbling strategies in polynomial time by a simple counting argument. However, perhaps somewhat surprisingly, as soon as we study *arbitrarily slowly* growing space, we can obtain superpolynomial trade-offs for formulas whose refutation space grows this slowly. This is a consequence of our new pebbling trade-off result in Section 5.2.

**Theorem 6.2.** *Let $g(n)$ be any arbitrarily slowly growing monotone function $\omega(1) = g(n) = \mathrm{O}\big(n^{1/7}\big)$, and let $\epsilon > 0$ be an arbitrarily small positive constant. Then there are explicitly constructible families of minimally unsatisfiable $k$-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

1. *Every formula $F_n$ is refutable in resolution in length $L(F_n \vdash 0) = \mathrm{O}(n)$ and also in variable space $\mathit{VarSp}(F_n \vdash 0) = \mathrm{O}(g(n))$ (but not simultaneously).*
2. *There are refutations $\pi_n : F_n \vdash 0$ in simultaneous variable space $\mathit{VarSp}(\pi_n) = \mathrm{O}\left(\sqrt[3]{n/g^2(n)}\right)$ and length $L(\pi_n) = \mathrm{O}(n)$.*
3. *There is a constant $K > 0$ such that any resolution refutation $\pi_n : F_n \vdash 0$ in clause space $Sp(\pi_n) \leq K\big(n/g^2(n)\big)^{1/3-\epsilon}$ must have length $L(\pi_n)$ superpolynomial in $n$.*

*The constant $K$ as well as the constants hidden in the asymptotic notation are independent of $n$ (but depend on $g$ and $\epsilon$).*

We remark that the upper-bound condition $g(n) = \mathrm{O}\big(n^{1/7}\big)$ is very mild and is there only for technical reasons in this theorem. If we allow the minimal space to grow as fast as $n^{\epsilon}$ for some $\epsilon > 0$, then there are other pebbling trade-off results that can give even stronger results for resolution than the one stated above (see, for instance, Section 6.4). Thus the interesting part is that $g(n)$ is allowed to grow arbitrarily slowly.

*Proof of Theorem 6.2.* Consider the graphs $\Gamma(c, r)$ in Definition 5.16. We want to choose the parameters $c$ and $r$ in a suitable way so that get a family of graphs in size $n = \Theta\big(cr^3 + c^3r^2\big)$ (using the bound on the size of $\Gamma(c, r)$ from Lemma 5.17). If we set

$$r = r(n) = g(n) \tag{49}$$

for $g(n) = \mathrm{O}\big(n^{1/7}\big)$, this forces

$$c = c(n) = \Theta\left(\sqrt[3]{n/g^2(n)}\right) \ . \tag{50}$$

Consider the graph family $\{G_n\}_{n=1}^{\infty}$ defined by $G_n = \Gamma(c(n), r(n))$ as in (49) and (50), which is a family of size $\Theta(n)$. Construct the single-sink version $\widehat{G_n}$ of $G_n$, fix any any non-authoritarian function $f$, consider the pebbling formulas $F_n = Peb_{\widehat{G_n}}[f]$, and appeal to the translation between pebbling and resolution in Theorem 4.4.

Lemma 5.17 yields that $VarSp(F_n \vdash 0) = \mathrm{O}(g(n))$. Also, the persistent black pebbling of $G_n$ in Lemma 5.18 yields a linear-time refutation $\pi_n : F_n \vdash 0$ with $VarSp(\pi_n) = \mathrm{O}\big(\sqrt[3]{n/g^2(n)}\big)$.

Now set the parameter $s$ in Theorem 5.19 to $s = c^{1-\epsilon'}$ for $\epsilon' = 3\epsilon$. Then for large enough $n$ we have $s \leq c/8 - 1$ and Theorem 5.19 can be applied. Combining the pebbling trade-off there with Theorem 4.4, we get that if the clause space is less than $\big(n/g^2(n)\big)^{1/3-\epsilon}$, then the required length of the refutation grows as $\big(\Omega\big(c\epsilon'\big)\big)^r = \big(\Omega\big(n/g^2(n)\big)\big)^{\epsilon g(n)}$ which is superpolynomial in $n$ for any $g(n) = \omega(1)$. The theorem follows. ☐

## 6.3 Robust Superpolynomial Trade-offs

We now know that there are polynomial trade-offs in resolution for constant space, and that going ever so slightly above constant space we can get superpolynomial trade-offs. The next question we want to focus on is how robust trade-offs we can get. That is, over how large a range of space does the trade-off hold? Given minimal refutation space $s$, how much larger space is needed in order to obtain the linear length refutation that we know exists for any pebbling contradiction?

The answer is that we can get superpolynomial trade-offs that span almost the whole range between constant and linear space. We present two different results illustrating this.

**Theorem 6.3.** *There are explicitly constructible families $\{F_n\}_{n=1}^{\infty}$ of minimally unsatisfiable $k$-CNF formulas of size $\Theta(n)$ such that:*

1. *Every formula $F_n$ is refutable in length $L(F_n \vdash 0) = \mathrm{O}(n)$ and variable space $VarSp(F_n \vdash 0) = \mathrm{O}(\log n)$, but not simultaneously.*
2. *There is a resolution refutation $\pi_n : F_n \vdash 0$ in variable space $VarSp(\pi_n) = \mathrm{O}\left(\sqrt[3]{n/\log^2 n}\right)$ and length $L(\pi_n) = \mathrm{O}(n)$.*
3. *There is a constant $K > 0$ such that any resolution refutation $\pi_n : F_n \vdash 0$ in clause space $Sp(\pi_n) \leq K\sqrt[3]{n/\log^2 n}$ must have length $L(\pi_n) = n^{\Omega(\log\log n)}$.*

*The constant $K$ as well as the constants hidden in the asymptotic notation are independent of $n$.*

*Proof.* Consider the graphs $\Gamma(c, r)$ in Definition 5.16 with parameters chosen so that $c = 2^r$. Then the size of $\Gamma(c, r)$ is $\Theta(r^2 2^{3r})$ by Lemma 5.17. Let $r(n) = \max\{r : r^2 2^{3r} \leq n\}$ and define the graph family $\{G_n\}_{n=1}^{\infty}$ by $G_n = \Gamma(2^r, r)$ for $r = r(n)$. Finally, construct the single-sink version $\widehat{G_n}$ of $G_n$, fix any any non-authoritarian function $f$ and consider the pebbling formulas $F_n = \mathit{Peb}_{\widehat{G_n}}[f]$ with the help of Theorem 4.4.

Translating from $G_n$ back to $\Gamma(c, r)$ we have parameters $r = \Theta(\log n)$ and $c = \Theta\big((n/\log^2 n)^{1/3}\big)$, so Lemma 5.17 yields that $\mathit{VarSp}(F_n \vdash 0) = \mathrm{O}(\log n)$. Also, the persistent black pebbling of $G_n$ in Lemma 5.18 yields a linear-time refutation $\pi_n : F_n \vdash 0$ with $\mathit{VarSp}(\pi_n) = \mathrm{O}\big((n/\log^2 n)^{1/3}\big)$.

Setting $s = c/8 - 1$ in Theorem 5.19 shows that there is a constant $K$ such that if the clause space of a refutation $\pi_n : F_n \vdash 0$ drops below $K \cdot (n/\log^2 n)^{1/3} \leq (r + 2) + s$, then we must have

$$L(\pi_n) \geq \mathrm{O}(1)^r \cdot r! = n^{\Omega(\log \log n)} \tag{51}$$

(where we used that $r = \Theta(\log n)$ for the final equality). The theorem follows. □

Sacrificing a square at the lower end of the interval, we can improve the upper end to $n/\log n$.

**Theorem 6.4.** *There are explicitly constructible families $\{F_n\}_{n=1}^{\infty}$ of minimally unsatisfiable $k$-CNF formulas of size $\Theta(n)$ such that:*

1. *Every formula $F_n$ is refutable in resolution in length $L(F_n \vdash 0) = \mathrm{O}(n)$ and also in variable space $\mathit{VarSp}(F_n \vdash 0) = \mathrm{O}(\log^2 n)$.*
2. *There is a resolution refutation $\pi_n : F_n \vdash 0$ in variable space $\mathit{VarSp}(\pi_n) = \mathrm{O}(n/\log n)$ and length $L(\pi_n) = \mathrm{O}(n)$.*
3. *There is a constant $K > 0$ such that any resolution refutation $\pi_n : F_n \vdash 0$ in clause space $\mathit{Sp}(\pi_n) \leq Kn/\log n$ must have length $L(\pi_n) = n^{\Omega(\log \log n)}$.*

*The constant $K$ and the constants hidden in the asymptotic notation are independent of $n$.*

*Proof.* Pick any non-authoritarian function $f$ and consider the pebbling formulas $\mathit{Peb}_{\widehat{\Phi(m,r)}}[f]$ defined over single-sink versions of stacks of superconcentrators $\Phi(m, r)$ as in Definition 5.33 with $m = 20T$ and $r = \lfloor n/T \rfloor$ for $T = \Theta(n/\log n)$. The theorem now follows by combining Theorem 5.34 with Theorem 4.4. □

We remark that the results in Theorem 6.4 can perhaps be considered to be slightly stronger than those in Theorem 6.3, but they require a very much more involved graph construction with worse hidden constants than the very simple and clean construction underlying Theorem 6.3.

## 6.4 Exponential Trade-offs

Superpolynomial trade-offs are all fine and well, but can we get *exponential* trade-offs? In this final subsection we answer this question in the affirmative.

The same counting argument that was mentioned in the beginning of Section 6.2 tells us that we can never expect to get exponential trade-offs from DAGs with polylogarithmic pebbling price. However, if we move to graphs with pebbling price $\Omega(n^{\epsilon})$ for some constant $\epsilon > 0$, pebbling formulas over such graphs can exhibit exponential trade-offs.

We obtain our first such exponential trade-off, which also exhibits a certain robustness, by again studying the DAGs in Definition 5.16.

**Theorem 6.5.** *There are explicitly constructible families $\{F_n\}_{n=1}^{\infty}$ of minimally unsatisfiable $k$-CNF formulas of size $\Theta(n)$ such that:*

1. *Every formula $F_n$ is refutable in resolution in length $L(F_n \vdash 0) = \mathrm{O}(n)$ and also in variable space $\mathit{VarSp}(F_n \vdash 0) = \mathrm{O}\big(\sqrt[8]{n}\big)$.*

2. *There is a resolution refutation $\pi_n : F_n \vdash 0$ in variable space $\mathit{VarSp}(\pi_n) = \mathrm{O}\big(\sqrt[4]{n}\big)$ and length $L(\pi_n) = \mathrm{O}(n)$.*

3. *There is a constant $K > 0$ such that any resolution refutation $\pi_n : F_n \vdash 0$ in clause space $\mathit{Sp}(\pi_n) \leq K\sqrt[4]{n}$ must have length $L(\pi_n) = \big(\sqrt[8]{n}\big)!$.*

*The constant $K$ as well as the constants hidden in the asymptotic notation are independent of $n$.*

*Proof.* Combine Theorem 4.4 and Theorem 5.19 in the same way as in the other proofs above for $\Gamma(c, r)$ with $c = \sqrt[4]{n}$ and $r = \sqrt[8]{n}$. □

We remark that there is nothing magic in our particular choice of parameters $c$ and $r$ in Theorem 6.5. Other parameters could be plugged in instead and yield slightly different results.

Now that we know that there are robust exponential trade-offs for resolution, we want to obtain exponential trade-offs for formulas with their minimal refutation space being as large as possible.

The higher the lower bound on space is, the more interesting the trade-off gets. It seems reasonable that to look at and analyze a CNF formula, a SAT solver will at some point use at least linear space. If so, it is not immediate to argue why the SAT solver would later work hard on optimizing lower order terms in the memory consumption and thus get stuck in a trade-off for relatively small space. Ideally, therefore, we would like to obtain trade-offs for superlinear space (if there are such trade-offs, that is). For such formulas, we would be more confident that the trade-off phenomena should also show up in practice.[7]

It is clear that pebbling contradictions can never yield any trade-off results in the superlinear regime, since they are always refutable in linear length and linear space simultaneously. Also, all trade-offs obtainable from the graphs in Definition 5.16 will be for space far below linear. However, using results from Section 5.3.3 we can get exponential trade-offs for space almost linear, or more precisely for space as large as $\Theta(n/\log n)$.

**Theorem 6.6.** *There are explicitly constructible families $\{F_n\}_{n=1}^{\infty}$ of minimally unsatisfiable $k$-CNF formulas of size $\Theta(n)$ such that:*

1. *Every formula $F_n$ is refutable in length $L(F_n \vdash 0) = \mathrm{O}(n)$ and variable space $\mathit{VarSp}(F_n \vdash 0) = \mathrm{O}(n/\log n)$.*

2. *There is a resolution refutation $\pi_n : F_n \vdash 0$ in variable space $\mathit{VarSp}(\pi_n) = \mathrm{O}(n)$ and length $L(\pi) = \mathrm{O}(n)$.*

3. *There is a constant $K > 0$ such that any resolution refutation $\pi_n : F_n \vdash 0$ in clause space $\mathit{Sp}(\pi_n) \leq Kn/\log n$, where $Kn/\log n \geq \mathit{Sp}(F_n \vdash 0)$, must have length $L(\pi) = \exp\big(n^{\epsilon}\big)$.*

*All constants, including those hidden in the asymptotic notation, are independent of $n$.*

*Proof.* Appeal to Theorem 5.36 in combination with Theorem 4.4 in the same way as in previous proofs in this section. □

[7] Having said that, we also want to point out that the case can certainly be made that even sublinear space trade-offs might be very relevant for real life applications. Intriguingly enough, pebbling contradictions over pyramids might in fact be an example of this. We know that these formulas have short, simple refutations, but in [SBK04] it was shown that state-of-the-art clause learning algorithms can have serious problems with even moderately large pebbling contradictions. (Their “grid pebbling formulas” are exactly our pebbling contradictions using substitution with binary, non-exclusive or.) We wonder whether the high lower bound on clause space can be part of the explanation behind this phenomenon.

We remark again that Theorem 5.36 in combination with Theorem 5.35 can be used to obtain DAGs (and thus CNF formulas) with other trade-offs as well for different space parameters in the range between $n/\log n$ and $n$. For simplicity and conciseness, however, we only state the special case above.

# 7 Directions for Further Research

We end by briefly mentioning a few open questions related to our reported work that we find most interesting.

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

**Open Question 1.** *Are there polynomial-size $k$-CNF formulas which require variable refutation space $\mathit{VarSp}(F \vdash 0) = \Omega\big((\textit{size of } F)^2\big)$?*

The answer has been conjectured by [ABSRW02] to be "yes", but as far as we are aware, there are no stronger lower bounds on variable space known than those that follow trivially from corresponding linear lower bounds on clause space. Thus, a first step would be to show superlinear lower bounds on variable space.

One way of interpreting the results of the current paper is that time-space trade-offs in pebble games carry over more or less directly to the resolution proof system (modulo the technical restrictions discussed in Section 4). The resolution trade-off results obtainable by this method are inherently limited, however, in the sense that pebblings in small space can be seen never to take too much time by a simple counting argument. For resolution there are no such limitations, at least not a priori, since the corresponding counting argument does not apply. Thus, one can ask whether it is possible to demonstrate even more dramatic time-space trade-offs for resolution than those obtained via pebbling.

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

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

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

We find Open Question 3 especially intriguing. Note that all bounds on clause space proven so far, inlcuding the trade-offs in the current paper, are in the regime where the space is less than formula size (which is quite natural, since by [ET01] we know the size of the formula is an upper bound on the minimal clause space needed). It is unclear to what extent such lower bounds on space are relevant to state-of-the-art SAT solvers, however, since such algorithms will presumably use at least a linear amount of memory to store the formula to begin with. For this reason, it seems to be a highly interesting problem to determine what can be said if we allow extra clause space above linear. Are there formulas exhibiting trade-offs in this superlinear regime, or is it always possible to carry out a minimal-length refutation in, say, at most a constant factor times the linear upper bound on the space required for any formula?

As was noted above, pebbling formulas cannot help answer these two questions, since pebbling formulas are always refutable in linear time and linear space simultaneously by construction, and since constant pebbling space implies polynomial pebbling time.

Finally, it would be interesting to investigate the implications of our results for applied satisfiability algorithms.

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

## Acknowledgements

The authors want to thank David Carlson, John Gilbert, Nicholas Pippenger, and John Savage, for helpful correspondence regarding their papers on pebbling.

## References

[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.

[AC03] Noga Alon and Michael Capalbo. Smaller explicit superconcentrators. *Internet Mathematics*, 1(2):151–163, 2003.

[AD03] Albert Atserias and Víctor Dalmau. A combinatorical characterization of resolution width. In *Proceedings of the 18th IEEE Annual Conference on Computational Complexity (CCC '03)*, pages 239–247, July 2003.

[AJPU02] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 448–456, May 2002.

[BEGJ00] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, 2000.

[BKPS02] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. The efficiency of resolution and Davis-Putnam procedures. *SIAM Journal on Computing*, 31(4):1048–1075, 2002.

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

[BOP03] Josh Buresh-Oppenheim and Toniann Pitassi. The complexity of resolution refinements. In *Proceedings of the 18th IEEE Symposium on Logic in Computer Science (LICS '03)*, pages 138–147, June 2003.

[Bor93] Allan Borodin. Time space tradeoffs (getting closer to the barrier?). In *Proceedings of the 4th International Symposium on Algorithms and Computation (ISAAC '93)*, pages 209–220, December 1993.

[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.

[BSIW04] Eli Ben-Sasson, Russell Impagliazzo, and Avi Wigderson. Near optimal separation of treelike and general resolution. *Combinatorica*, 24(4):585–603, September 2004.

[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. A space hierarchy for $k$-DNF resolution. Manuscript in preparation, 2009.

[BSSV03] Paul Beame, Michael Saks, Xiaodong Sun, and Erik Vee. Time-space trade-off lower bounds for randomized computation of decision problems. *Journal of the ACM*, 50(2):154–195, March 2003.

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

[Coo74] Stephen A. Cook. An observation on time-storage trade off. *Journal of Computer and System Sciences*, 9:308–316, 1974.

[CS76] Stephen A. Cook and Ravi Sethi. Storage requirements for deterministic polynomial time recognizable languages. *Journal of Computer and System Sciences*, 13(1):25–37, 1976.

[CS80] David A. Carlson and John E. Savage. Graph pebbling with many free pebbles can be difficult. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC '80)*, pages 326–332, 1980.

[CS82] David A. Carlson and John E. Savage. Extreme time-space tradeoffs for graphs with small space requirements. *Information Processing Letters*, 14(5):223–227, 1982.

[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.

[ET03] Juan Luis Esteban and Jacobo Torán. A combinatorial characterization of treelike resolution space. *Information Processing Letters*, 87(6):295–300, 2003.

[FLvMV05] Lance Fortnow, Richard Lipton, Dieter van Melkebeek, and Anastasios Viglas. Time-space lower bounds for satisfiability. *Journal of the ACM*, 52(6):835–865, November 2005.

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, 1981.

[GLT80] John R. Gilbert, Thomas Lengauer, and Robert Endre Tarjan. The pebbling problem is complete in polynomial space. *SIAM Journal on Computing*, 9(3):513–524, August 1980.

[GT78] John R. Gilbert and Robert Endre Tarjan. Variations of a pebble game on graphs. Technical Report STAN-CS-78-661, Stanford University, 1978. Available at the webpage `http://infolab.stanford.edu/TR/CS-TR-78-661.html`.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[Her08] Alex Hertel. *Applications of Games to Propositional Proof Complexity*. PhD thesis, University of Toronto, May 2008. Available at `http://www.cs.utoronto.ca/~ahertel/`.

[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.

[HPV77] John Hopcroft, Wolfgang Paul, and Leslie Valiant. On time versus space. *Journal of the ACM*, 24(2):332–337, April 1977.

[Kla85] Maria M. Klawe. A tight bound for black and white pebbles on the pyramid. *Journal of the ACM*, 32(1):218–228, January 1985.

[KS88] Balasubramanian Kalyanasundaram and George Schnitger. On the power of white pebbles. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, pages 258–266, 1988.

[LT80] Thomas Lengauer and Robert Endre Tarjan. The space complexity of pebble games on trees. *Information Processing Letters*, 10(4/5):184–188, July 1980.

[LT82] Thomas Lengauer and Robert Endre Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *Journal of the ACM*, 29(4):1087–1130, October 1982.

[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.

[Pip77] Nicholas Pippenger. Superconcentrators. *SIAM Journal on Computing*, 6(2):298–304, June 1977.

[Pip80] Nicholas Pippenger. Pebbling. Technical Report RC8258, IBM Watson Research Center, 1980. Appeared in Proceedings of the 5th IBM Symposium on Mathematical Foundations of Computer Science, Japan.

[PTC77] Wolfgang J. Paul, Robert Endre Tarjan, and James R. Celoni. Space bounds for a game on graphs. *Mathematical Systems Theory*, 10:239–251, 1977.

[Pud97] Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *Journal of Symbolic Logic*, 62(3):981–998, September 1997.

[Raz03] Alexander A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theoretical Computer Science*, 1(303):233–243, June 2003.

[Raz04] Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *Journal of the ACM*, 51(2):115–138, 2004.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, March 1999.

[Rob65] John Alan Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[SBK04] Ashish Sabharwal, Paul Beame, and Henry Kautz. Using problem structure for efficient clause learning. In *6th International Conference on Theory and Applications of Satisfiability Testing (SAT ’03), Selected Revised Papers*, volume 2919 of *Lecture Notes in Computer Science*, pages 242–256. Springer, 2004.

[Seg07] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):482–537, December 2007.

[Tor99] Jacobo Torán. Lower bounds for space in resolution. In *Proceedings of the 13th International Workshop on Computer Science Logic (CSL ’99)*, volume 1683 of *Lecture Notes in Computer Science*, pages 362–373. Springer, 1999.

[Tor04] Jacobo Torán. Space and width in propositional resolution. *Bulletin of the European Association for Theoretical Computer Science*, 83:86–104, June 2004.

[Tse68] Grigori Tseitin. On the complexity of derivation in propositional calculus. In A. O. Silenko, editor, *Structures in Constructive Mathematics and Mathematical Logic, Part II*, pages 115–125. Consultants Bureau, New York-London, 1968.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

[Val76] Leslie G. Valiant. Graph-theoretic properties in computational complexity. *Journal of Computer and System Sciences*, 13(3):278–285, 1976.

[vM06] Dieter van Melkebeek. A survey of lower bounds for satisfiability and related problems. *Foundations and Trends in Theoretical Computer Science*, 2(3):197–303, January 2006.

[Wil85] Robert Wilber. White pebbles help. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC ’85)*, pages 103–112, 1985.



ECCC ISSN 1433-8092
http://eccc.hpi-web.de/