Electronic Colloquium on Computational Complexity, Report No. 2 (2009)

Short Proofs May Be Spacious: An Optimal Separation of Space and Length in Resolution

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

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

November 23, 2008

Abstract

A number of works have looked at the relationship between length and space of resolution proofs. A notorious question has been whether the existence of a short proof implies the existence of a proof that can be verified using limited space.

In this paper we resolve the question by answering it negatively in the strongest possible way. We show that there are families of 6-CNF formulas of size n, for arbitrarily large n, that have resolution proofs of length $\mathrm{O}(n)$ but for which any proof requires space $\Omega(n/\log n)$. This is the strongest asymptotic separation possible since any proof of length $\mathrm{O}(n)$ can always be transformed into a proof in space $\mathrm{O}(n/\log n)$.

Our result follows by reducing the space complexity of so called pebbling formulas over a directed acyclic graph to the black-white pebbling price of the graph. The proof is somewhat simpler than previous results (in particular, those reported in [Nordström 2006, Nordström and Håstad 2008]) as it uses a slightly different flavor of pebbling formulas which allows for a rather straightforward reduction of proof space to standard black-white pebbling price.

1 Introduction

Resolution length and space Perhaps the single most studied proof system in propositional proof complexity is *resolution*. This system made its first appearance in 1937 in [Bla37] and began to be investigated in connection with automated theorem proving in the 1960s [DLL62, DP60, Rob65]. Because of the simplicity of resolution—there is only one derivation rule—and because all lines in a proof are clauses, this proof system readily lends itself to proof search algorithms.

*Research supported in part by an Alon Fellowship and grants by the Israeli Science Foundation and by the US-Israel Binational Science Foundation.

†Research supported in part by the Ericsson Research Foundation, the Foundation Olle Engkvist Byggma"stare, and the Foundation Blanceflor Boncompagni-Ludovisi, ne'e Bildt.

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

ISSN 1433-8092

Being so simple and fundamental, resolution was also a natural target to attack when developing methods for proving lower bounds in proof complexity. In this context, it is most straightforward to prove bounds on the *length* of refutations, i.e., the number of clauses, rather than on the total size of refutations. The length and size measures are easily seen to be polynomially related. In 1968, Tseitin [Tse68] presented a superpolynomial lower bound on refutation length for a restricted form of resolution, called *regular* resolution, but it was not until almost 20 years later that Haken [Hak85] proved the first superpolynomial lower bound for general resolution. This weakly exponential bound of Haken has later been followed by many other strong results, among others truly exponential lower bounds on resolution refutation length for different formula families in, for instance, [BKPS02, BSW01, CS88, Urq87].

The formal study of *space* in resolution was initiated by Esteban and Torán [ET01, Tor99]. Intuitively, the space of a resolution refutation is the maximal number of clauses one needs to keep in memory while verifying the refutation, and the space of refuting the CNF formula F is defined as the minimal space of any resolution refutation of F. A number of upper and lower bounds for refutation space in resolution and other proof systems have subsequently been presented in, for example, [ABSRW02, BSG03, EGM04, ET03].

With the definition of space complexity, a natural question to ask is how space relates to other complexity measures of propositional proofs. Esteban and Torán [ET01] proved that the space is at most logarithmic in the minimal length of a treelike refutation of a formula, which implies that space is bounded by the number of variables appearing in the formula. The question of the relation between space and length of general resolution proofs, which is the focus of this paper, was raised by the first author in [BS02] and has also been discussed in, for instance, [ET03, Seg07, Tor04], but there has been no consensus on what the right answer should be. However, these papers identify a plausible formula family for answering the question, namely so-called *pebbling contradictions* defined in terms of pebble games over directed acyclic graphs (DAGs) and these formulas have indeed been used in [Nor06, NH08] to make progress and, in this paper, finally resolve the question.

While understanding the relation between space and length seemed stuck, progress was reported on another front — that of space versus *width*. The width measure, first made explicit by Galil in [Gal77], is defined as the maximal number of literals in a clause in the refutation. Atserias and Dalmau showed in [AD03] that space is always greater than width, raising the possibility of equivalence of these two measures. Notice that width is a different measure of "proof space" as it is the maximal "space" occupied by a single line in the refutation and one may have speculated that the two "space" measures are in fact equivalent.

Progress on the space-length question for general resolution was finally obtained by the second author in [Nor06], which also separated space from width. This was done by exhibiting a k-CNF formula family of size $\mathrm{O}(n)$ refutable in width $\mathrm{O}(1)$ and length $\mathrm{O}(n)$ but requiring space $\Theta(\log n)$. In a recent joint work of the second author with Håstad [NH08] this separation was improved to width $\mathrm{O}(1)$ and length $\mathrm{O}(n)$ versus space $\Theta(\sqrt{n})$ for a related formula family. We note however that this previous state-of-the-art did not rule out the existence of a space-length tradeoff quantitatively similar to the width-length tradeoff of [BSW01] which says width is at most $\mathrm{O}(\sqrt{n \cdot \text{ length }})$.

Our contribution In this paper, we finally resolve the open question about the relationship between space and length by establishing an optimal separation between the two measures. We do this by studying a somewhat modified variant of pebbling contradictions defined using XORs (see Definition 3.1) and proving lower bounds for such *XOR-pebbling contradictions* in terms of the pebbling price of the underlying DAGs.

Theorem 1.1 (Main). *The space of refuting XOR-pebbling contradictions over any DAG G in resolution is lower-bounded by the black-white pebbling price of G, provided that the number of variables per vertex in the XOR-pebbling contradictions is at least* 2.

If we take a constant number of variables per vertex and study DAGs with constant fan-in, it is easy to show that XOR-pebbling contradictions can be refuted in linear length and constant width. Using the result from [GT78] which exhibits a family of fan-in 2 DAGs $\{G_n\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ having pebbling price $\Omega(n/\log n)$, we get the following corollary.

Corollary 1.2 (Main). *There is a family $\{F_n\}_{n=1}^{\infty}$ of 6-CNF formulas of size $\mathrm{O}(n)$ that can be refuted in length $\mathrm{O}(n)$ and width $\mathrm{O}(1)$ but require space $\Omega(n/\log n)$.*

Since it can be proven using results from [ET01, HPV77] that a refutation of length $\mathrm{O}(n)$ can be carried out in space $\mathrm{O}(n/\log n)$, the separation of space and length in Corollary 1.2 is asymptotically optimal. As an extra bonus, we note that while the constructions in [Nor06, NH08] are quite intricate and the proofs very involved, our optimal lower bound proof is relatively clean and straightforward and we discuss it next.

Proof outline For the purposes of analyzing space, a resolution derivation from a CNF formula F can be viewed as a sequence of derivation steps on a blackboard. In each step we may write a clause from F on the blackboard (an *axiom* clause), erase a clause from the blackboard or derive some new clause implied by the clauses currently written on the blackboard. The space of a derivation is then the maximum number of clauses on the blackboard simultaneously.

The black-white pebble game models non-deterministic computation, and the black-white pebbling price of a DAG G is the minimal number of memory registers needed to verify the calculation described by G, where the source vertices contain the input and non-source vertices specify operations on the values of the predecessors. The pebble game on a DAG G can be encoded as an unsatisfiable CNF formula, a so-called *pebbling contradiction* over G.

Pebble games have been used extensively as a tool to prove time and space lower bounds and tradeoffs for computation. Loosely put, a lower bound for the pebbling price of a graph says that although the computation that the graph describes can be performed quickly, it requires large space. Our hope is that when we encode pebble games in terms of CNF formulas, these formulas should inherit the same properties as the underlying graphs. That is, if we pick a DAG G with high pebbling price, since the corresponding pebbling contradiction encodes a calculation which needs a lot of memory we would like to try to argue that any resolution refutation of this formula should require large space.

Ideally, we would like to give a proof of a lower bound on the resolution refutation space of pebbling contradictions along the following lines:

1. First, find a natural interpretation of sets of clauses currently "on the blackboard" in a refutation of the pebbling contradiction over G in terms of black and white pebbles on the vertices of the DAG G.

2. Then, prove that this interpretation captures the pebble game in the following sense: for any resolution refutation of a pebbling contradiction over G, looking at consecutive sets of clauses on the blackboard and considering the corresponding sets of pebbles we get a black-white pebbling of G.

3. Finally, show that the interpretation captures clause space in the sense that if the content of the blackboard induces N pebbles on the graph, then there must be at least N clauses on the blackboard.

Combining the above with known lower bounds on the pebbling price of G, this would imply a lower bound on the refutation space of pebbling contradictions. The separation from length and width would then follow since pebbling contradictions are known to be refutable in linear length and constant width.

Unfortunately, this idea does not quite work "off the shelf." Pebblings of DAGs and resolution refutations of CNF formulas are very different objects, and there is no reason a priori that there should be a tight connection between the two. However, relaxing the requirements for the correspondence between resolution and pebbling, the papers [Nor06, NH08] made essentially the proof idea above work for two special cases of graphs. In this paper, by using related ideas and studying a slightly modified variant of pebbling contradictions, we can handle any graph, which results in an optimal separation of space and length.

Implications for practical SAT-solvers In recent years, SATISFIABILITY has gone from a problem of mainly theoretical interest to a practical approach for solving applied problems. Although all known Boolean satisfiability solvers (SAT-solvers) have exponential running time in the worst case, enormous progress in performance has led to satisfiability algorithms becoming a standard tool for solving a large number of real-world problems such as hardware and software verification, experiment design, and scheduling.

Perhaps a somewhat surprising aspect of this development is that the most successful SAT-solvers to date are still variants of the resolution-based Davis-Putnam-Logemann-Loveland (DPLL) procedure [DLL62, DP60] augmented with *clause learning*. For instance, the great majority of the best algorithms at the 2007 round of the international SAT competitions [SAT] fit this description. DPLL procedures perform a recursive backtrack search in the space of partial truth value assignments. The idea behind clause learning, or *conflict-driven learning*, is that at each failure (backtrack) point in the search tree, the system derives a reason for the inconsistency in the form of a new clause and then adds this clause to the original CNF formula ("learning" the clause). This can save a lot of work later on in the proof search, when some other partial truth value assignment fails for similar reasons. The main bottleneck for this approach, other than the obvious one of time, is the amount of memory used by the algorithms. Thus, understanding time and memory requirements for clause learning algorithms, and how these requirements are related to one another is a question of great practical importance. We refer to, e.g., [BKS03, KS07, Sab05] for a more detailed discussion of clause learning (and SAT-solving in general) with examples of applications.

In the field of proof complexity, the resources of time and memory correspond to the length and space of resolution proofs. Our work indicates that on certain input formulas, a short proof does not necessarily imply a space-efficient proof exists. Let us give one implication of our result to questions regarding the practical construction of DPLL-based SAT-solvers.

Consider a "frugal" DPLL-based solver augmented with clause learning that tries to save memory by limiting the number of learned clauses as a function of its running time. The reasoning underlying the frugal algorithm is very natural — to save running time, start with the very minimal possible resources and increase them slowly as necessary. Appealing as this strategy may seem, our work shows that on certain inputs it will perform much worse than other, more prodigal, strategies.[1]

Organization of the rest of the paper After stating the necessary definitions in Section 2, we state and discuss our main results in Section 3. Section 4 defines the "resolution-pebbling game" that we use as an intermediate step when translating resolution refutations into black-white pebblings. In Sections 5–7 we provide the proof of our main theorem. Section 8 contains some short concluding remarks.

2 Preliminaries

For the sake of completeness, before presenting our main results we briefly recount (verbatim) from [Nor08] a few basic definitions regarding resolution and pebble games that will be used later on.

[1]This issue is somewhat subtle, however, and out of space considerations we cannot give a full discussion here. Let us just note that there are empirical results like [SBK04] indicating that although pebbling contradictions have very short resolution proofs, these proofs can be very hard to find even for a state-of-the-art SAT-solver.

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* $C = a_1 \vee \cdots \vee a_k$ 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* $F = C_1 \wedge \cdots \wedge C_m$ 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, α, β truth value assignments and ν 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 α everywhere except possibly at x, to which it assigns the value ν. We let $Vars(C)$ denote the set of variables and $Lit(C)$ 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 $[n] = \{1, 2, \ldots, n\}$.

A *resolution derivation* $\pi : F \vdash A$ of a clause A from a CNF formula F is 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 (*axioms*) or is derived from clauses D_j, D_k in π with $j, k < i$ by the *resolution rule*

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

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. A *resolution refutation* of a CNF formula F is a resolution derivation of the empty clause 0, i.e., the clause with no literals, from F. (Perhaps somewhat confusingly, this is sometimes also referred to as a *resolution proof* of F.)

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 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.

With every resolution derivation $\pi : F \vdash A$ we can associate a DAG G_π, with the clauses in π labeling the vertices and with edges from the assumption clauses to the resolvent for each application of the resolution rule (2). There might be several different derivations of a clause C in π, but if so we can label each occurrence of C with a timestamp when it was derived and keep track of which copy of C is used where. A resolution derivation π is *tree-like* if any clause in the derivation is used at most once as a premise in an application of the resolution rule, i.e., if G_π is a tree. (We may make different "time-stamped" vertex copies of the axiom clauses in order to make G_π into a tree).

The *length* $L(\pi)$ of a resolution derivation π is the number of clauses in it. We define the length of deriving a clause A from a formula F as $L(F \vdash A) = \min_{\pi : F \vdash A}\{L(\pi)\}$, where the minimum is taken over all resolution derivations of A. In particular, the length of refuting F by resolution is denoted $L(F \vdash 0)$. The

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

length of refuting F by tree-like resolution $L_{\mathfrak{T}}(F \vdash 0)$ is defined by taking the minimum over all tree-like resolution refutations π_T of F.

The *width* $W(C)$ of a clause C is $|C|$, i.e., the number of literals appearing in it. The width of a set of clauses $\mathbb{C}$ is $W(\mathbb{C}) = \max_{C \in \mathbb{C}}\{W(C)\}$. The width of deriving A from F by resolution is $W(F \vdash A) = \min_{\pi : F \vdash A}\{W(\pi)\}$, and the width of refuting F is denoted $W(F \vdash 0)$. Note that the minimum width measures in general and tree-like resolution coincide, so it makes no sense to make a separate definition for $W_{\mathfrak{T}}(F \vdash 0)$.

We next define the measure of *space*. Following the exposition in [ET01], a proof can be seen 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. The *clause space* of a resolution proof is the maximum number of clauses that need to be kept in memory simultaneously during a verification of the proof. For the formal definition, it is convenient to use the alternative definition of resolution introduced in [ABSRW02].

Definition 2.1 (Resolution). 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[3] of the following rules:

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

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* of F is a derivation of the empty clause 0 from F.

Definition 2.2 (Clause space [ABSRW02, BS02]). The *clause space* of a resolution derivation $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is $\max_{t \in [\tau]}\{|\mathbb{C}_t|\}$. The clause space of deriving A from F is $\mathit{Sp}(F \vdash A) = \min_{\pi : F \vdash A}\{\mathit{Sp}(\pi)\}$, and $\mathit{Sp}(F \vdash 0)$ denotes the minimum clause space of any resolution refutation of F.

Restricting the resolution derivations to tree-like resolution, we get the measure $\mathit{Sp}_{\mathfrak{T}}(F \vdash 0)$ in analogy with $L_{\mathfrak{T}}(F \vdash 0)$ defined above.

We note that for technical reasons, it is sometimes convenient to add a rule for *weakening*, 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 resolution refutation without increasing the length, width or space.

A technical tools that we will use to simplify some of the proofs are *restrictions*.

Definition 2.3 (Restriction). A *partial assignment* or *restriction* ρ is a partial function $\rho : X \mapsto \{0, 1\}$, where X is a set of Boolean variables. We identify ρ with the set of literals $\{a_1, \ldots, a_m\}$ set to true by ρ. The *ρ-restriction* of a clause C is defined to be

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

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., $\rho(\neg C) = \{\overline{a} \mid a \in C\}$.

[3] In some previous papers, resolution is defined so as to allow every derivation step to *combine* one or zero applications of each of the three derivation rules. Therefore, some of the bounds stated in this paper for space as defined next are off by a constant as compared to the cited sources.

Proposition 2.4. *If π is a resolution refutation of F and ρ 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 and space as π.*

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 an easy induction over the resolution refutation π.

2.2 The Black-White Pebble Game

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]). An excellent survey of pebbling up to ca 1980 is [Pip80].

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 2.5 (Black-white pebble game). Suppose that G is a DAG with sources S and a unique target 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_1, W_1) to (B_2, W_2) in G is a sequence of pebble configurations $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = (B_1, W_1)$, $\mathbb{P}_\tau = (B_2, W_2)$, and for all $t \in [\tau]$, $\mathbb{P}_t$ follows from $\mathbb{P}_{t-1}$ by one of the rules above. If $(B_1, W_1) = (\emptyset, \emptyset)$, we say that the pebbling is *unconditional*, otherwise it is *conditional*.

The *cost* of a pebble configuration $\mathbb{P} = (B, W)$ is $\mathit{cost}(\mathbb{P}) = |B \cup W|$ and the cost of a pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is $\max_{0 \leq t \leq \tau}\{\mathit{cost}(\mathbb{P}_t)\}$. The *black-white pebbling price* of (B, W), denoted $\mathit{BW\text{-}Peb}(B, W)$, is the minimum cost of any unconditional pebbling reaching (B, W).

A *complete pebbling* of G, also called a *pebbling strategy* for G, is an unconditional pebbling reaching $(\{z\}, \emptyset)$. The *black-white pebbling price* of G, denoted $\mathit{BW\text{-}Peb}(G)$, is the minimum cost of any complete black-white pebbling of G.

3 Main Results

In this section we formally present and discuss our main results. To this end we start by defining the class of formulas that we analyze in this paper.

Pebbling Formulas Let $\oplus_{i=1}^{d} x_i$ denote the xor of $x_1, \ldots, x_d$ and $\overline{\oplus}_{i=1}^{d} x_i$ denote the negation of this formula. The satisfying assignments of $\oplus_{i=1}^{d} x_i$ ($\overline{\oplus}_{i=1}^{d} x_i$, respectively) are assignments with an odd (even, respectively) number of 1's. In what follows, we associate a Boolean formula with the CNF formula that is logically equivalent to it in the canonical way. For instance, the formula $(x \oplus y) \rightarrow (z \oplus w)$, which is equivalent to $(x \overline{\oplus} y) \vee (z \oplus w)$, is associated with the CNF formula

$$(\overline{x} \vee y \vee z \vee w) \wedge (\overline{x} \vee y \vee \overline{z} \vee \overline{w}) \wedge (x \vee \overline{y} \vee z \vee w) \wedge (x \vee \overline{y} \vee \overline{z} \vee \overline{w}) \ .$$

The next definition is a generalization of formulas previously studied in [BSW01, BEGJ00, RM99].

Definition 3.1 (XOR-pebbling contradiction). Let G be a DAG with sources S, a unique sink z, and let $d > 0$ be an integer. Associate d distinct variables $v_1, \ldots, v_d$ with every vertex $v \in V(G)$. The dth degree *XOR-pebbling contradiction* over G, denoted $Peb_G^d[\oplus]$, is the CNF obtained from the conjunction of the following formulas over xor-constraints:

- **Source Axioms:** $\bigoplus_{i=1}^{d} s_i$ for all sources $s \in S$.
- **Pebbling Axioms:** For all vertices $u^{(1)}, \ldots, u^{(\ell)}, v$, such that $u^{(1)}, \ldots, u^{(\ell)}$ are all the immediate predecessors of v, we have $\bigoplus_{i=1}^{d} u_i^{(1)} \wedge \ldots \wedge \bigoplus_{i=1}^{d} u_i^{(\ell)} \rightarrow \bigoplus_{i=1}^{d} v_i$, which is equivalent to the disjunction
$$\overline{\bigoplus}_{i=1}^{d} u_i^{(1)} \vee \ldots \vee \overline{\bigoplus}_{i=1}^{d} u_i^{(\ell)} \vee \bigoplus_{i=1}^{d} v_i.$$
- **Sink Axioms:** $\overline{\bigoplus}_{i=1}^{d} z_i$ for the sink z.

See Figure 1 on the facing page for an example XOR-pebbling contradiction.

If G has n vertices and maximal in-degree ℓ, then $Peb_G^d[\oplus]$ is an unsatisfiable $(\ell + 1)d$-CNF formula with at most $2^{(\ell+1)(d-1)} \cdot n$ clauses over $d \cdot n$ variables.

We can now give a more precise statement of our lower bound on refutation space for XOR-pebbling contradictions.

Theorem 1.1 (restated). *For every $d > 1$, there is a constant c such that for any DAG G it holds that*

$$Sp(Peb_G^d[\oplus] \vdash 0) \geq \textit{\textsf{BW-Peb}}(G) - c \ .$$

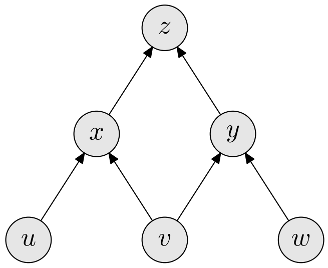
In what follows, a family of formulas $\{F_n\}_{n=1}^{\infty}$ is said to be *explicitly constructible* if there exists a polynomial time Turing machine that on input 1^n outputs F_n.

Corollary 1.2 (restated). *For every $d > 1$, there is a explicitly constructible family $\{F_n\}_{n=1}^{\infty}$ of $3d$-CNF formulas of size $\mathrm{O}(n)$ such that $L(F_n \vdash 0) = \mathrm{O}(n)$ and $W(F_n \vdash 0) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Omega(n/\log n)$.*

Proof. For any DAG G with n vertices, in-degree 2 and a single sink, the CNF formula $Peb_G^2[\oplus]$ is a $3d$-CNF of size $\mathrm{O}(n)$ that can be refuted using proofs of length $\mathrm{O}(n)$ and width $\mathrm{O}(1)$ (for a proof see [BS02, Theorem 4.3]). The lower bound on space and the explicit constructibility of the formulas follow respectively from Theorem 1.1 and the following lower bound on black-white pebbling price. ☐

Theorem 3.2 ([GT78]). *There is a family of explicitly constructible[4] DAGs G_n with $\Theta(n)$ vertices and vertex indegree 2 for all non-sources such that $\textit{\textsf{BW-Peb}}(G) = \Theta(n/\log n)$.*

[4]This was not known at the time of the original theorem in [GT78]. What is needed is an explicit construction of superconcentrators of linear density, and it has since been shown in [GG81] how to do this with [AC03] presenting the currently best construction.



$$
\begin{array}{ll}
(u_1 \lor u_2) & \land\, (v_1 \lor \overline{v}_2 \lor \overline{w}_1 \lor w_2 \lor y_1 \lor y_2) \\
\land\, (\overline{u}_1 \lor \overline{u}_2) & \land\, (v_1 \lor \overline{v}_2 \lor \overline{w}_1 \lor w_2 \lor \overline{y}_1 \lor \overline{y}_2) \\
\land\, (v_1 \lor v_2) & \land\, (\overline{v}_1 \lor v_2 \lor w_1 \lor \overline{w}_2 \lor y_1 \lor y_2) \\
\land\, (\overline{v}_1 \lor \overline{v}_2) & \land\, (\overline{v}_1 \lor v_2 \lor w_1 \lor \overline{w}_2 \lor \overline{y}_1 \lor \overline{y}_2) \\
\land\, (w_1 \lor w_2) & \land\, (\overline{v}_1 \lor v_2 \lor \overline{w}_1 \lor w_2 \lor y_1 \lor y_2) \\
\land\, (\overline{w}_1 \lor \overline{w}_2) & \land\, (\overline{v}_1 \lor v_2 \lor \overline{w}_1 \lor w_2 \lor \overline{y}_1 \lor \overline{y}_2) \\
\land\, (u_1 \lor \overline{u}_2 \lor v_1 \lor \overline{v}_2 \lor x_1 \lor x_2) & \land\, (x_1 \lor \overline{x}_2 \lor y_1 \lor \overline{y}_2 \lor z_1 \lor z_2) \\
\land\, (u_1 \lor \overline{u}_2 \lor v_1 \lor \overline{v}_2 \lor \overline{x}_1 \lor \overline{x}_2) & \land\, (x_1 \lor \overline{x}_2 \lor y_1 \lor \overline{y}_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\, (u_1 \lor \overline{u}_2 \lor \overline{v}_1 \lor v_2 \lor x_1 \lor x_2) & \land\, (x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor y_2 \lor z_1 \lor z_2) \\
\land\, (u_1 \lor \overline{u}_2 \lor \overline{v}_1 \lor v_2 \lor \overline{x}_1 \lor \overline{x}_2) & \land\, (x_1 \lor \overline{x}_2 \lor \overline{y}_1 \lor y_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\, (\overline{u}_1 \lor u_2 \lor v_1 \lor \overline{v}_2 \lor x_1 \lor x_2) & \land\, (\overline{x}_1 \lor x_2 \lor y_1 \lor \overline{y}_2 \lor z_1 \lor z_2) \\
\land\, (\overline{u}_1 \lor u_2 \lor v_1 \lor \overline{v}_2 \lor \overline{x}_1 \lor \overline{x}_2) & \land\, (\overline{x}_1 \lor x_2 \lor y_1 \lor \overline{y}_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\, (\overline{u}_1 \lor u_2 \lor \overline{v}_1 \lor v_2 \lor x_1 \lor x_2) & \land\, (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor y_2 \lor z_1 \lor z_2) \\
\land\, (\overline{u}_1 \lor u_2 \lor \overline{v}_1 \lor v_2 \lor \overline{x}_1 \lor \overline{x}_2) & \land\, (\overline{x}_1 \lor x_2 \lor \overline{y}_1 \lor y_2 \lor \overline{z}_1 \lor \overline{z}_2) \\
\land\, (v_1 \lor \overline{v}_2 \lor w_1 \lor \overline{w}_2 \lor y_1 \lor y_2) & \land\, z_1 \lor \overline{z}_2 \\
\land\, (v_1 \lor \overline{v}_2 \lor w_1 \lor \overline{w}_2 \lor \overline{y}_1 \lor \overline{y}_2) & \land\, \overline{z}_1 \lor z_2
\end{array}
$$

Figure 1: The XOR-pebbling contradiction $Peb^2_{\Pi_2}[\oplus]$ for the pyramid graph Π_2 of height 2.

Proof of Theorem 1.1. There are three main components to our proof of Theorem 1.1. In the next section we define and discuss the *resolution-pebbling price* of a DAG G, denoted $\textsf{\textit{Res-Peb}}(G)$. Then we prove the following pair of statements. The first theorem is proved in Sections 5,6 and the second is proved in Section 7. Taken together, they complete the proof of Theorem 1.1. $\square$

Theorem 3.3. *For every $d > 1$, there is a constant c such that for any DAG G it holds that*

$$Sp(Peb^d_G[\oplus] \vdash 0) \geq \textsf{\textit{Res-Peb}}(G) - c \ .$$

Theorem 3.4. *For any DAG G it holds that*

$$\textsf{\textit{Res-Peb}}(G) \geq \textsf{\textit{BW-Peb}}(G) \ .$$

4 The Resolution-Pebbling Game

In this section we define our modified pebble game that will be used to analyze resolution refutations. The next definition is similar to [NH08], but somewhat simpler.

Definition 4.1 (Res-pebbling subconfiguration). If B and W are sets of vertices in a DAG G with $B \neq \emptyset$, $B \cap W = \emptyset$, we say that $[B]\langle W\rangle$ is a *res-pebbling subconfiguration*, or just *subconfiguration*, in G with black pebbles on B and white pebbles on W *supporting* B. A set of subconfigurations $\mathbb{R} = \{[B_i]\langle W_i\rangle \,|\, i = 1, \ldots, m\}$ is a *res-pebbling configuration* and its *cost* is $\textsf{\textit{cost}}(\mathbb{R}) = |\bigcup_{i=1}^m (B_i \cup W_i)|$.

The game that we play with subconfigurations is also similar to that in [NH08], although noticeably less complicated.

Definition 4.2 (Resolution-pebbling game). For G a DAG, a *resolution-pebbling*, or *res-pebbling* for short, is a sequence $\mathcal{R} = \{\mathbb{R}_0, \ldots, \mathbb{R}_\tau\}$ of pebbling clause configurations such that for every $t \in [\tau]$, the configuration $\mathbb{R}_t$ is obtained from $\mathbb{R}_{t-1}$ by one of the following rules:

Download $\mathbb{R}_t = \mathbb{R}_{t-1} \cup \{[v]\langle pred(v)\rangle\}$, where $pred(v)$ denotes the set of predecessors of v. (Notice $pred(v) = \emptyset$ for a source node v.)

Resolution $\mathbb{R}_t = \mathbb{R}_{t-1} \cup \{[B_1 \cup B_2]\langle W_1 \cup W_2\rangle\}$ if there exist $[B_1]\langle W_1 \cup \{v\}\rangle$ and $[B_2 \cup \{v\}]\langle W_2\rangle$ in $\mathbb{R}_{t-1}$ such that $B_1 \cap W_2 = \emptyset$.

Weakening $\mathbb{R}_t = \mathbb{R}_{t-1} \cup \{[B \cup B']\langle W \cup W'\rangle\}$ if $[B]\langle W\rangle \in \mathbb{R}_{t-1}$ and $(B \cup B') \cap (W \cup W') = \emptyset$.

Erasure $\mathbb{R}_t = \mathbb{R}_{t-1} \setminus \{[B]\langle W\rangle\}$ for $[B]\langle W\rangle \in \mathbb{R}_{t-1}$.

The cost of a resolution-pebbling is $\mathit{cost}(\mathcal{R}) = \max_{t\in[\tau]}\{\mathit{cost}(\mathbb{R}_t)\}$. The *resolution-pebbling price* of G is the minimal cost of a resolution pebbling starting with $\mathbb{R}_0 = \emptyset$ and ending with $\mathbb{R}_\tau = \{[z]\langle\emptyset\rangle\}$ where z is the sink of G:

$$\mathit{Res\text{-}Peb}(G) = \min\{\mathit{cost}(\mathcal{R}) \mid \mathcal{R} = \{\mathbb{R}_0, \ldots, \mathbb{R}_\tau\} \text{ with } \mathbb{R}_0 = \emptyset \text{ and } \mathbb{R}_\tau = \{[z]\langle\emptyset\rangle\}\}.$$

Let us try to provide some intuition for the pebbling rules. We interpret a subconfiguration $[B]\langle W\rangle$ as saying "If all vertices in W have a white pebble on them, then a black pebble can be placed somewhere in B via a legal sequence of black-white pebbling moves." A res-pebbling configuration is a set of such statements and the res-pebbling game is a system that allows for deducing new true statements from existing ones. Indeed, going over the four allowed moves in Definition 4.2 one can verify that they give rise to legal statements. For instance, a download step allows us to state: "If all predecessors of v have a white pebble, then a black pebble may be placed on v." The case of the Resolution rule is perhaps the most subtle so we will describe it in detail. The pair *(i)* $[B_1]\langle W_1 \cup \{v\}\rangle$, *(ii)* $[B_2 \cup \{v\}]\langle W_2\rangle$ says: *(i)* "If white pebbles are placed on $W_1 \cup \{v\}$ we may place a black pebble somewhere in B_2", and *(ii)* "If white pebbles are placed on W_2 we may place a black pebble somewhere on $B_2 \cup \{v\}$". The new statement derived by a resolution step says: *(iii)* "If $W_1 \cup W_2$ are covered by white pebbles then a black pebble may be placed somewhere on $B_1 \cup B_2$." Indeed, if all of $W_1 \cup W_2$ have white pebbles, then by statement *(ii)* we know a black pebble may be placed somewhere on $B_2 \cup \{v\}$. If it is placed in B_2 we are done because *(iii)* is true. Otherwise, the black pebble is placed on v. Then by statement *(i)* a black pebble may be placed somewhere on B_1 after which the black pebble can be removed from v. This shows why, intuitively, the resolution step should be valid. The cases of weakening and erasure can be argued in a similar fashion.

5 Resolution Derivations Induce Res-Pebblings

The proof of Theorem 3.3 follows from two main steps. The first step argues that every refutation π of $Peb^d_G[\oplus]$ induces a res-pebbling $\mathcal{R}_\pi$. The second step says that the cost of the induced res-pebbling $\mathcal{R}_\pi$ is a lower bound on the space of π. Together, these two steps imply Theorem 3.3.

In this section, we do the first step by showing how resolution derivations can be interpreted in terms of resolution-pebblings. As in [Nor06, NH08], we get a cleaner correspondence between resolution and pebbling if we ignore the sink axioms $\overline{\bigoplus}_{i=1}^{d} z_i$ and instead study resolution derivations of $\bigoplus_{i=1}^{d} z_i$ from the rest of the formula rather than refutations of all of $Peb^d_G[\oplus]$. Let us write $*Peb^d_G[\oplus] = Peb^d_G[\oplus] \setminus \{\overline{\bigoplus}_{i=1}^{d} z_i\}$ to denote the pebbling formula over G with the sink axioms in the pebbling contradiction removed. The next lemma is the formal statement saying that as long as we keep the pebbling degree d constant, we may just as well study resolution derivations of $\bigoplus_{i=1}^{d} z_i$ from $*Peb^d_G[\oplus]$ instead of refutations of $Peb^d_G[\oplus]$ without losing more than a constant term. The proof, which is similar to [Nor06, NH08], is omitted due to space constraints.

Lemma 5.1. *For any DAG G with sink z, it holds that* $Sp(Peb^d_G[\oplus] \vdash 0) = Sp\big({*}Peb^d_G[\oplus] \vdash \bigoplus_{i=1}^d z_i\big) + \mathrm{O}\big(2^d\big)$.

Proof. For any resolution derivation $\pi^* : {*}Peb^d_G[\oplus] \vdash \bigoplus_{i=1}^d z_i$, we can get a refutation of $Peb^d_G[\oplus]$ from π^* in at most $\mathrm{O}\big(2^d\big)$ extra space by downloading all sink axioms defining $\overline{\bigoplus}_{i=1}^d z_i$ and then, keeping all clauses in memory, deriving the empty clause in additional space $d + \mathrm{O}(1)$ (since any formula over d variables is refutable in space $d + \mathrm{O}(1)$ by [ET01]).

In the other direction, suppose we have a refutation $\pi : Peb^d_G[\oplus] \vdash 0$. Let ρ be a partial assignment to $z_1, \ldots, z_d$ such that $\overline{\bigoplus}_{i=1}^d \rho(z_i)$. Consider the restricted refutation $\pi\restriction_\rho$. This restriction satisfies all sink axioms, so these axioms are never used in the restricted refutation $\pi\restriction_\rho$. Also, it is not hard to see that the restricted refutation has space at most $Sp(\pi)$. Removing the restriction again, we get a resolution derivation $\pi_C : {*}Peb^d_G[\oplus] \vdash C$ where C is the unique clause over $z_1, \ldots, z_d$ that is not satisfied by ρ. Notice $C \vDash (\bigoplus_{i=1}^d z_i)$. Ranging over all 2^{d-1} partial assignments ρ satisfying $\overline{\bigoplus}_{i=1}^d \rho(z_i)$ we derive all clauses implying $\bigoplus_{i=1}^d z_i$. Keeping all such clauses in memory we conclude the overall space required to derive $\bigoplus_{i=1}^d z_i$ is at most $Sp(\pi) + 2^{d-1}$. ☐

In view of Lemma 5.1, from now on we will only consider resolution derivations from ${*}Peb^d_G[\oplus]$ and translate clause configurations in such derivations into sets of black and white pebbles. Note that since ${*}Peb^d_G[\oplus]$ is non-contradictory and resolution is sound, any clause set $\mathbb{C}$ derived from ${*}Peb^d_G[\oplus]$ is satisfiable. We next specify how to translate clauses to pebbles.

Definition 5.2 (Induced res-pebbling subconfiguration). Let G be a DAG and $\mathbb{C}$ a set of clauses derived from ${*}Peb^d_G[\oplus]$. Then $\mathbb{C}$ induces the res-pebbling subconfiguration $[B]\langle W\rangle$ if

$$\mathbb{C} \vDash \big(\bigvee_{b\in B} \bigoplus_{i=1}^d b_i\big) \vee \big(\bigvee_{w\in W} \overline{\bigoplus}_{i=1}^d w_i\big) \tag{3a}$$

but for all strict subsets $B' \subsetneqq B$ and $W' \subsetneqq W$ that

$$\mathbb{C} \nvDash \big(\bigvee_{b\in B'} \bigoplus_{i=1}^d b_i\big) \vee \big(\bigvee_{w\in W} \overline{\bigoplus}_{i=1}^d w_i\big) \ \text{, and} \tag{3b}$$

$$\mathbb{C} \nvDash \big(\bigvee_{b\in B} \bigoplus_{i=1}^d b_i\big) \vee \big(\bigvee_{w\in W'} \overline{\bigoplus}_{i=1}^d w_i\big) \ . \tag{3c}$$

To save space, when all conditions (3a)–(3c) hold, we write

$$\mathbb{C} \rhd \big(\bigvee_{b\in B} \bigoplus_{i=1}^d b_i\big) \vee \big(\bigvee_{w\in W} \overline{\bigoplus}_{i=1}^d w_i\big) \tag{4}$$

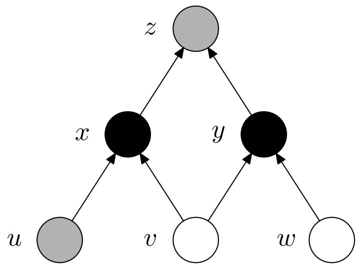
and refer to this as *precise implication*. We also say that the clause set $\mathbb{C}$ implies $\big(\bigvee_{b\in B} \bigoplus_{i=1}^d b_i\big) \vee \big(\bigvee_{w\in W} \overline{\bigoplus}_{i=1}^d w_i\big)$ *precisely*. We will also overload the notation and write $\mathbb{C} \vDash [B]\langle W\rangle$, $\mathbb{C} \nvDash [B]\langle W\rangle$, and $\mathbb{C} \rhd [B]\langle W\rangle$ when the corresponding implications or non-implications hold for $\mathbb{C}$ with respect to $\big(\bigvee_{b\in B} \bigoplus_{i=1}^d b_i\big) \vee \big(\bigvee_{w\in W'} \overline{\bigoplus}_{i=1}^d w_i\big)$. We write

$$\mathbb{R}(\mathbb{C}) = \big\{[B]\langle W\rangle \big| \mathbb{C} \rhd [B]\langle W\rangle\big\} \tag{5}$$

to denote the set of all res-pebbling subconfigurations induced by $\mathbb{C}$.

$$\begin{bmatrix} x_1 \lor x_2 \\ \overline{x}_1 \lor \overline{x}_2 \\ v_1 \lor w_1 \lor y_1 \lor y_2 \\ v_1 \lor w_1 \lor \overline{y}_1 \lor \overline{y}_2 \\ v_2 \lor w_1 \lor y_1 \lor y_2 \\ v_2 \lor w_1 \lor \overline{y}_1 \lor \overline{y}_2 \\ v_1 \lor w_2 \lor y_1 \lor y_2 \\ v_1 \lor w_2 \lor \overline{y}_1 \lor \overline{y}_2 \\ v_2 \lor w_2 \lor y_1 \lor y_2 \\ v_2 \lor w_2 \lor \overline{y}_1 \lor \overline{y}_2 \end{bmatrix}$$

(a) Clauses on blackboard.



(b) Corresponding pebbles in the graph.

Figure 2: Example of correspondence between clauses and pebbles for XOR-pebbling contradiction.

See Figure 2 for an example of how clauses are translated into pebbles in this way.

The following theorem forms the first part of the proof of Theorem 3.3 and says that resolution derivations induce legal res-pebbling sequences.

Theorem 5.3. *Let* $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ *be a resolution derivation of* $\bigoplus_{i=1}^{d} z_i$ *from* $^*Peb_G^d[\oplus]$. *Then the induced res-pebbling configurations* $\{\mathbb{R}(\mathbb{C}_0), \ldots, \mathbb{R}(\mathbb{C}_\tau)\}$ *form the "backbone" of a complete res-pebbling* $\mathcal{R}$ *of* G *in the sense that*

1. $\mathbb{R}(\mathbb{C}_0) = \emptyset$,
2. $\mathbb{R}(\mathbb{C}_\tau) = \{[z]\langle\emptyset\rangle\}$, *and*
3. *for every* $t \in [\tau]$, *the transition* $\mathbb{R}(\mathbb{C}_{t-1}) \rightsquigarrow \mathbb{R}(\mathbb{C}_t)$ *can be accomplished in accordance with the res-pebbling rules in cost* $\max\{\mathit{cost}(\mathbb{R}(\mathbb{C}_{t-1})), \mathit{cost}(\mathbb{R}(\mathbb{C}_t))\} + \mathrm{O}(1)$.

In particular, to any resolution derivation $\pi : {}^*Peb_G^d[\oplus] \vdash \bigoplus_{i=1}^{d} z_i$ *we can associate a complete res-pebbling* $\mathcal{R}_\pi$ *of* G *such that* $\mathit{cost}(\mathcal{R}_\pi) \leq \max_{\mathbb{C}\in\pi}\{\mathit{cost}(\mathbb{R}(\mathbb{C}))\} + \mathrm{O}(1)$.

Before proving Theorem 5.3 let us try to describe in words what the theorem says. Using the translation of clauses into pebbles in Definition 5.2, clause configurations $\mathbb{C}_0, \mathbb{C}_1, \ldots, \mathbb{C}_\tau$ in a resolution derivation π can be seen to correspond to "snapshots" at different time intervals of a res-pebbling $\mathcal{R}_\pi$ of the DAG G. Furthermore, the cost of this pebbling is essentially upper-bounded by the largest cost we see at any of the snapshots. There may be many pebbling moves needed to go from the pebble configuration corresponding to $\mathbb{C}_{t-1}$ to the one corresponding to $\mathbb{C}_t$, but the maximal cost during this intermediate pebbling moves is at most an additive constant larger than the cost of the pebble configuration corresponding to $\mathbb{C}_{t-1}$ or $\mathbb{C}_t$. Later on we use this to show that the cost of the res-pebbling $\mathcal{R}_\pi$ yields a lower bound on the space of the resolution refutation π.

Proof of Theorem 5.3. Property 1 above follows from $\mathbb{C}_0 = \emptyset$ and property 2 follows from $\mathbb{C}_\tau = \{\bigoplus_{i=1}^{d} z_i\}$. Thus, we focus on proving property 3. Since the resolution-pebbling game allows us to erase any res-pebbling subconfiguration, we only have to show that every new subconfiguration at time t

$$[B]\langle W\rangle \in \mathbb{R}(\mathbb{C}_t) \setminus \mathbb{R}(\mathbb{C}_{t-1}) \tag{6}$$

can be obtained by res-pebbling moves starting with $\mathbb{R}(\mathbb{C}_{t-1})$ and that the intermediate res-pebbling configurations in between $\mathbb{R}(\mathbb{C}_{t-1})$ and $\mathbb{R}(\mathbb{C}_t)$ do not affect the pebbling cost by more than an additive constant. Let us first take care of the two easy cases.

If $\mathbb{C}_t$ follows from $\mathbb{C}_{t-1}$ by an erasure, then $\mathbb{R}(\mathbb{C}_t)\backslash\mathbb{R}(\mathbb{C}_{t-1}) = \emptyset$ as is easily verified from Definition 5.2. Thus, the only thing that can happen is that subconfigurations disappear, and we can get from $\mathbb{R}(\mathbb{C}_{t-1})$ to $\mathbb{R}(\mathbb{C}_t)$ by performing the corresponding erasure moves in the res-pebbling. This decreases the pebbling cost monotonically.

If $\mathbb{C}_t$ follows from $\mathbb{C}_{t-1}$ by an inference, then no subconfigurations can disappear. Furthermore, we have that $\mathbb{C}_{t-1} \vDash \mathbb{C}_t$, which implies by (3a) that every subconfiguration $[B]\langle W\rangle \in \mathbb{R}(\mathbb{C}_t)$ satisfies $\mathbb{C}_{t-1} \vDash [B]\langle W\rangle$. But then it is straightforward to show that all new subconfigurations $[B]\langle W\rangle$ can be derived from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening moves. For completeness, we write down this as a formal statement and provide a proof.

Observation 5.4. *Suppose that $\mathbb{C} \vDash \left(\bigvee_{b\in B} \bigoplus_{i=1}^{d} b_i\right) \vee \left(\bigvee_{w\in W} \overline{\bigoplus}_{i=1}^{d} w_i\right)$ for a clause set $\mathbb{C}$ derived from $*Peb_G^d[\oplus]$. Then there is a subconfiguration $[B']\langle W'\rangle \in \mathbb{R}(\mathbb{C})$ such that $B' \subseteq B$ and $W' \subseteq W$. In particular, $[B]\langle W\rangle$ is derivable by weakening from $\mathbb{R}(\mathbb{C})$.*

Proof. Just pick any minimal clause set $\mathbb{C}' \subseteq \mathbb{C}$, and any minimal vertex sets $B' \subseteq B$ and $W' \subseteq W$ such that the implication $\mathbb{C}' \vDash \left(\bigvee_{b\in B'} \bigoplus_{i=1}^{d} b_i\right) \vee \left(\bigvee_{w\in W'} \overline{\bigoplus}_{i=1}^{d} w_i\right)$ holds. (We note that $B' \neq \emptyset$ since $*Peb_G^d[\oplus] \nvDash \bigvee_{w\in W} \left(\oplus_{i=1}^{d} w_i = 0\right)$ and resolution is sound.) But then by Definition 5.2, it holds that $\mathbb{C}' \rhd \left(\bigvee_{b\in B'} \bigoplus_{i=1}^{d} b_i\right) \vee \left(\bigvee_{w\in W'} \overline{\bigoplus}_{i=1}^{d} w_i\right)$, so $[B']\langle W'\rangle \in \mathbb{R}(\mathbb{C})$ and clearly $[B]\langle W\rangle$ can be derived from $[B']\langle W'\rangle$ by weakening. $\square$

In particular, this means that the pebbling cost increases monotonically when going from $\mathbb{R}(\mathbb{C}_{t-1})$ to $\mathbb{R}(\mathbb{C}_t)$ if the derivation step at time t is an inference.

The interesting case is when the derivation step at time t is an axiom download, i.e., $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{C\}$ for some axiom C. This is more complicated, and we will spend the rest of this section showing how to take care of this case.

First, we need some notation. Recall that we identify the constraints $\overline{\bigoplus}_{i=1}^{d} x_i$ and $\bigoplus_{i=1}^{d} x_i$ with the canonical CNF formulas over $x_1, \ldots, x_d$ which are logically equivalent to these constraints. That is, we interpret $\overline{\bigoplus}_{i=1}^{d} x_i$ and $\bigoplus_{i=1}^{d} x_i$ as sets of clauses. For convenience of notation, we also define the disjunction $\mathbb{C} \vee \mathbb{D}$ of two clause sets $\mathbb{C}$ and $\mathbb{D}$ to be the clause set

$$\mathbb{C} \vee \mathbb{D} = \{C \vee D \mid C \in \mathbb{C},\, D \in \mathbb{D}\} \ . \tag{7}$$

This notation extends to more than two clause sets in the natural way.

If r is a non-source vertex with predecessors $pred(r) = \{p, q\}$, we say that the *axioms for* r in $*Peb_G^d[\oplus]$ are

$$Ax^d(r) = \overline{\bigoplus}_{i=1}^{d} p_i \vee \overline{\bigoplus}_{i=1}^{d} q_i \vee \bigoplus_{i=1}^{d} r_i \tag{8}$$

where using the notational convention in (7) we have that $\overline{\bigoplus}_{i=1}^{d} p_i \vee \overline{\bigoplus}_{i=1}^{d} q_i \vee \bigoplus_{i=1}^{d} r_i$ is the set of clauses

$$\left\{C_{\neg p} \vee C_{\neg q} \vee C_r \middle| C_{\neg p} \in \overline{\bigoplus}_{i=1}^{d} p_i,\, C_{\neg q} \in \overline{\bigoplus}_{i=1}^{d} q_i,\, C_r \in \bigoplus_{i=1}^{d} r_i\right\} \ , \tag{9}$$

and if r is a source, we define

$$Ax^d(r) = \bigoplus_{i=1}^{d} r_i \ . \tag{10}$$

For U a set of vertices in G, we let $Ax^d(U) = \bigcup_{u\in U} Ax^d(u)$. Note that with this notation, we have $*Peb_G^d[\oplus] = \bigcup_{v\in V(G)} Ax^d(v)$.

A key tool in the proof that will follow is the next technical observation, which is an easy consequence of Observation 5.4 once has deciphered the notation.

Observation 5.5. *If $\mathbb{C}$ is a clause set derived from *$^*Peb^d_G[\oplus]$ *and $C \in Ax^d(r)$ is an axiom clause for some vertex $r \in V(G)$ such that*

$$\mathbb{C} \cup \{C\} \vDash \Big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\Big) \vee \Big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^{d} w_i\Big) \ , \tag{11}$$

then:

1. *It always holds that $\mathbb{C} \vDash \big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\big) \vee \big(\bigvee_{w \in W \cup \{r\}} \overline{\bigoplus}_{i=1}^{d} w_i\big)$, so if $r \notin B$ we can derive $[B]\langle W \cup \{r\}\rangle$ from $\mathbb{R}(\mathbb{C})$ by weakening.*

2. *If r is a non-source and $q \in pred(r)$, then $\mathbb{C} \vDash \big(\bigvee_{b \in B \cup \{q\}} \bigoplus_{i=1}^{d} b_i\big) \vee \big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^{d} w_i\big)$ holds. In particular, if $q \notin W$ we can derive $[B \cup \{q\}]\langle W\rangle$ from $\mathbb{R}(\mathbb{C})$ by weakening.*

Proof. If $C \in Ax^d(r)$, then by (8) and (10) there is a subclause $D \subseteq C$ such that $D \in \bigoplus_{i=1}^{d} r_i$. Suppose that α is any assignment with $\alpha(\mathbb{C}) = 1$ but $\alpha\big(\big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\big) \vee \big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^{d} w_i\big)\big) = 0$ (if there is no such α then we are already done). Then we must have $\alpha(C) = 0$ since otherwise we get a contradiction to (11), so in particular $\alpha(D) = 0$. But then $\alpha\big(\overline{\bigoplus}_{i=1}^{d} r_i\big) = 1$. Hence, any assignment α that satisfies $\mathbb{C}$ must also satisfy $\big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\big) \vee \big(\bigvee_{w \in W \cup \{r\}} \overline{\bigoplus}_{i=1}^{d} w_i\big)$. Applying Observation 5.4, we get part 1 above.

Part 2 is very similar. If $C \in Ax^d(r)$ for a non-source vertex r with $q \in pred(r)$, there is a subclause $D \subseteq C$ such that $D \in \overline{\bigoplus}_{i=1}^{d} q_i$ (compare (9) above). Let us again pick any truth value assignment α such that $\alpha(\mathbb{C}) = 1$ but $\alpha\big(\big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\big) \vee \big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^{d} w_i\big)\big) = 0$. Then it must hold that $\alpha(C) = 0$, but this implies that $\alpha(D) = 0$ and $\alpha\big(\bigoplus_{i=1}^{d} q_i\big) = 1$. ☐

Returning to our proof of Theorem 5.3 in the case when the derivation step at time t is a download of an axiom $C \in Ax^d(r)$, assume that this induces a new res-pebbling subconfiguration $[B]\langle W\rangle$. Then C must be one of the clauses inducing the subconfiguration, and we get that there is a clause set $\mathbb{C} \subseteq \mathbb{C}_{t-1}$ such that

$$\mathbb{C} \cup \{C\} \rhd \Big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\Big) \vee \Big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^{d} w_i\Big) \ . \tag{12}$$

Our intuition is that download of an axiom clause $C \in Ax^d(r)$ in the resolution derivation should correspond to an introduction of $[r]\langle pred(r)\rangle$ in the induced res-pebbling. We want to prove that any other res-pebbling subconfiguration $[B]\langle W\rangle$ in $\mathbb{R}(\mathbb{C}_t)$ is derivable by the pebbling rules from $\mathbb{R}(\mathbb{C}_{t-1}) \cup \big\{[r]\langle pred(r)\rangle\big\}$. We will also need to prove that the resolution-pebbling moves needed to go from $\mathbb{R}(\mathbb{C}_{t-1})$ to $\mathbb{R}(\mathbb{C}_t)$ do not increase the res-pebbling cost by more than an additive constant compared to

$$\max\big\{\textit{cost}(\mathbb{R}(\mathbb{C}_{t-1})), \textit{cost}(\mathbb{R}(\mathbb{C}_t))\big\} = \textit{cost}(\mathbb{R}(\mathbb{C}_t)),$$

where the equality holds since no subconfigurations induced by $\mathbb{C}_{t-1}$ can disappear when we add clauses to $\mathbb{C}_{t-1}$.

As a warm-up, let us consider the case when r is a source, i.e., $pred(r) = \emptyset$ and $C \in Ax^d(r) = \bigoplus_{i=1}^{d} r_i$. We make a case analysis depending on whether $r \in B$ in (12) or not.

1. $r \in B$: In this case we need no further analysis. Just make the res-pebbling download move $[r]\langle\emptyset\rangle$ and weaken $[r]\langle\emptyset\rangle$ to get $[B \cup \{r\}]\langle W\rangle = [B]\langle W\rangle$.

2. $r \notin B$: By part 1 of Observation 5.5, we can derive $[B]\langle W \cup \{r\}\rangle$ by weakening from $\mathbb{R}(\mathbb{C}_{t-1})$. Then $[B]\langle W\rangle$ can be derived by a download of $[r]\langle\emptyset\rangle$ followed by a resolution of $[B]\langle W \cup \{r\}\rangle$ and $[r]\langle\emptyset\rangle$.

We see that when r is a source, we can get from $\mathbb{R}(\mathbb{C}_{t-1})$ to $\mathbb{R}(\mathbb{C}_t)$ by a download of $[r]\langle\emptyset\rangle$ and possibly some weakenings and resolutions.

The case when r is a non-source is a bit more involved, but the general idea is the same. Suppose for the rest of this section that $C \in Ax^d(r)$ for some fixed vertex r with $pred(r) = \{p, q\}$. This means that C can be written $C = C_{\neg p} \vee C_{\neg q} \vee C_r$ for some $C_{\neg p} \in \overline{\bigoplus}_{i=1}^d p_i$, $C_{\neg q} \in \overline{\bigoplus}_{i=1}^d q_i$, and $C_r \in \bigoplus_{i=1}^d r_i$, and we can rewrite (12) as

$$\mathbb{C} \cup \{C_{\neg p} \vee C_{\neg q} \vee C_r\} \vartriangleright \Big(\bigvee_{b \in B} \bigoplus_{i=1}^d b_i\Big) \vee \Big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^d w_i\Big) \ . \tag{13}$$

Let us also assume that

$$\mathbb{C}_{t-1} \nvDash \Big(\bigvee_{b \in B} \bigoplus_{i=1}^d b_i\Big) \vee \Big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^d w_i\Big) \tag{14}$$

since otherwise we can derive $[B]\langle W\rangle$ by an weakening move from $\mathbb{R}(\mathbb{C}_{t-1})$ (using Observation 5.4) and be done. Recall that by definition, we have $B \cap W = \emptyset$. Observe that it must hold that

$$\{p, q\} \cap B = \emptyset \ , \tag{15}$$

since if, say, $q \in B$, we could apply part 1 of Observation 5.5 to get that the implication in (14) in fact holds for $B = B \cup \{q\}$ contrary to assumption. In the same way, we see that

$$r \notin W \tag{16}$$

since otherwise part 2 of Observation 5.5 shows that the implication (14) on the contrary is true for $W = W \cup \{r\}$.

As in the case when r was a source vertex, the induction step is by a case analysis depending on whether or not $r \in B$ in the implication (13) (which, we remind ourselves, is just (12) with added information about what the downloaded axiom clause C looks like).

1. $r \in B$: We split this case into subcases depending on whether $p, q \in W$ or not. By the symmetry of p and q, we have the following three possibilities to consider:

   (a) $\{p, q\} \subseteq W$,

   (b) $p \in W, q \notin W$,

   (c) $\{p, q\} \cap W = \emptyset$.

   We analyze these cases in order.

   (a) $\{p, q\} \subseteq W$: This is the easiest case. Since by assumption $r \in B$ and $\{p, q\} \subseteq W$, the subconfiguration $[B]\langle W\rangle \in \mathbb{R}(\mathbb{C}_t)$ can be derived by a download of $[\{r\}]\langle\{p, q\}\rangle$ followed by a weakening of $[\{r\}]\langle\{p, q\}\rangle$ to $[B \cup \{r\}]\langle W \cup \{p, q\}\rangle = [B]\langle W\rangle$.

   (b) $p \in W$, $q \notin W$: In this case $[\{r\}]\langle\{p, q\}\rangle$ cannot be weakened to $[B]\langle W\rangle$, since $q \notin W$. We need to find some way to eliminate the white pebble on q. But since $q \notin W$, part 2 of Observation 5.5 says that we can derive $[B \cup \{q\}]\langle W\rangle$ by weakening from $\mathbb{R}(\mathbb{C}_{t-1})$. Using this subconfiguration, we can derive $[B]\langle W\rangle$ as follows:

      - download $[\{r\}]\langle\{p, q\}\rangle$,
      - derive $[B \cup \{q\}]\langle W\rangle$ from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening,
      - resolve $[\{r\}]\langle\{p, q\}\rangle$ and $[B \cup \{q\}]\langle W\rangle$ to get $[B \cup \{r\}]\langle W \cup \{p\}\rangle = [B]\langle W\rangle$.

      Note that the resolution move is in accordance with the rules since $\{r\} \cap W = \emptyset$ as noted in (16) and $(B \cup \{q\}) \cap \{p, q\} = \{q\}$ as noted in (15).

(c) $\{p, q\} \cap W = \emptyset$: Now both p and q have to be eliminated if we are to use $[\{r\}]\langle\{p, q\}\rangle$ to derive $[B]\langle W\rangle$, but by applying part 2 of Observation 5.5 twice we see that we can derive $[B \cup \{p\}]\langle W\rangle$ and $[B \cup \{q\}]\langle W\rangle$ by weakening from $\mathbb{R}(\mathbb{C}_{t-1})$. Using this fact, we can perform a pebbling to get $[B]\langle W\rangle$ as follows:

- download $[\{r\}]\langle\{p, q\}\rangle$,
- derive $[B \cup \{q\}]\langle W\rangle$ from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening,
- resolve $[\{r\}]\langle\{p, q\}\rangle$ with $[B \cup \{q\}]\langle W\rangle$ on q to get $[B \cup \{r\}]\langle W \cup \{p\}\rangle = [B]\langle W \cup \{p\}\rangle$,
- derive $[B \cup \{p\}]\langle W\rangle$ from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening,
- conclude by resolving $[B]\langle W \cup \{p\}\rangle$ with $[B \cup \{p\}]\langle W\rangle$ on p, resulting in the subconfiguration $[B]\langle W\rangle$.

Of course, it needs to be checked that all resolution moves are legal, but this follows from (15) and (16).

This concludes the analysis for the case $r \in B$ for a non-source vertex r.

2. $r \notin B$: This case is quite similar to the previous case $r \in B$. Here also we make a subcase analysis depending on whether $|pred(r) \cap W|$ is equal to 2, 1 or 0.

Before we do this, though, we observe that there is a particular subconfiguration that will be useful for us. Since we are now assuming that $r \notin B$, part 1 of Observation 5.5 says that $[B]\langle W \cup \{r\}\rangle$ is derivable by weakening from $\mathbb{R}(\mathbb{C}_{t-1})$. This subconfiguration will play an important role in the pebblings below.

(a) $\{p, q\} \subseteq W$: To get the subconfiguration $[B]\langle W\rangle$ from $\mathbb{R}(\mathbb{C}_{t-1})$ in this case, first derive the subconfiguration $[B]\langle W \cup \{r\}\rangle$ just mentioned by weakening from $\mathbb{R}(\mathbb{C}_{t-1})$, then download $[\{r\}]\langle\{p, q\}\rangle$, and finally resolve the two to get $[B]\langle W \cup \{p, q\}\rangle = [B]\langle W\rangle$. This resolution move is in accordance with the res-pebbling rules since $B \cap \{p, q\} = \emptyset$ according to (15) and $\{r\} \cap (W \cup \{r\}) = \{r\}$.

(b) $p \in W$, $q \notin W$: Just as in case 1b, part 2 of Observation 5.5 says that $[B \cup \{q\}]\langle W\rangle$ is derivable from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening. Now do the following pebbling moves:

- download $[\{r\}]\langle\{p, q\}\rangle$,
- derive $[B \cup \{q\}]\langle W\rangle$ from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening using part 2 of Observation 5.5 as in case 1b,
- resolve $[\{r\}]\langle\{p, q\}\rangle$ with $[B \cup \{q\}]\langle W\rangle$ on q to get $[B \cup \{r\}]\langle W \cup \{p\}\rangle$,
- use part 1 of Observation 5.5 to derive $[B]\langle W \cup \{r\}\rangle$ from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening,
- finally, resolve $[B \cup \{r\}]\langle W \cup \{p\}\rangle$ with $[B]\langle W \cup \{r\}\rangle$ on r to get $[B]\langle W \cup \{p\}\rangle = [B]\langle W\rangle$.

(c) $\{p, q\} \cap W = \emptyset$: As in case 1c, appeal to part 2 of Observation 5.5 twice to find subconfigurations $[B \cup \{p\}]\langle W\rangle, [B \cup \{q\}]\langle W\rangle$ derivable from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening. Using that $[B]\langle W \cup \{r\}\rangle$ also can be derived from $\mathbb{R}(\mathbb{C}_{t-1})$ by weakening, we can make the following sequence of pebbling moves:

- download $[\{r\}]\langle\{p, q\}\rangle$,
- derive $[B \cup \{q\}]\langle W\rangle$ by weakening,
- resolve $[\{r\}]\langle\{p, q\}\rangle$ and $[B \cup \{q\}]\langle W\rangle$ on q to derive $[B \cup \{r\}]\langle W \cup \{p\}\rangle$,
- derive $[B \cup \{p\}]\langle W\rangle$ by weakening,
- resolve $[B \cup \{r\}]\langle W \cup \{p\}\rangle$ and $[B \cup \{p\}]\langle W\rangle$ on p to derive $[B \cup \{r\}]\langle W\rangle$,

- derive $[B]\langle W \cup \{r\}\rangle$ by weakening,
- finally, resolve $[B \cup \{r\}]\langle W\rangle$ and $[B]\langle W \cup \{r\}\rangle$ on r resulting in $[B]\langle W\rangle$.

Double-checking the set intersections and inclusions shows that all these moves are legal.

This concludes the analysis for the case $r \notin B$.

We are finally through the case analysis for axiom download. Let us put the bits and pieces together and argue why Theorem 5.3 now follows.

If $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is a derivation of $\bigoplus_{i=1}^{d} z_i$ from $*Peb_G^d[\oplus]$, it is easily verified from Definition 5.2 that $\mathbb{R}(\mathbb{C}_0) = \mathbb{R}(\emptyset) = \emptyset$ and $\mathbb{R}(\mathbb{C}_\tau) = \mathbb{R}(\bigoplus_{i=1}^{d} z_i) = \{[z]\langle\emptyset\rangle\}$. Above, we have shown how to do the intermediate res-pebbling moves to get from $\mathbb{R}(\mathbb{C}_{t-1})$ to $\mathbb{R}(\mathbb{C}_t)$ in the case of erasure, inference and axiom download, respectively. For erasure and inference, we already noted that the res-pebbling cost changes monotonically during the transition $\mathbb{R}(\mathbb{C}_{t-1}) \rightsquigarrow \mathbb{R}(\mathbb{C}_t)$. In the case of axiom download, all pebbles used in the intermediate moves are still on the DAG in $\mathbb{R}(\mathbb{C}_t)$ except possibly for the pebbles on $\{r\} \cup \mathit{pred}(r)$, so the extra intermediate cost is upper-bounded by 3. This shows that the complete res-pebbling $\mathcal{R}_\pi$ of the DAG G associated to any resolution derivation $\pi : *Peb_G^d[\oplus] \vdash \bigoplus_{i=1}^{d} z_i$ by the construction in this section has res-pebbling cost bounded from above by $\textsf{cost}(\mathcal{R}_\pi) \leq \max_{\mathbb{C}\in\pi}\{\textsf{cost}(\mathbb{R}(\mathbb{C}))\} + 3$. Theorem 5.3 is thereby proven. ☐

6 Comparing Resolution Space and Res-Pebbling Cost

In this section, we provide the second component in the proof of Theorem 3.3, namely, that the cost of the induced resolution pebbling $\mathcal{R}_\pi$ is a lower bound on the space of π.

We introduce some notation to make the argument more concise. Let us write $\mathit{Vars}^d(u) = \{u_1, \ldots, u_d\}$. We say that a vertex u is *represented* in a clause C derived from $*Peb_G^d[\oplus]$, or that C *mentions* u, if $\mathit{Vars}^d(u) \cap \mathit{Vars}(C) \neq \emptyset$. We write

$$V(C) = \{u \in V(G) \,|\, \mathit{Vars}^d(u) \cap \mathit{Vars}(C) \neq \emptyset\} \tag{17}$$

to denote all vertices represented in C. We will also refer to $V(C)$ as the set of vertices *mentioned* by C. This notation is extended to sets of clauses by taking unions.

The main component in the proof of Theorem 3.3 is the following theorem. We remark that this is the place in the proof where it is absolutely crucial that we are working with XOR-pebbling contradictions $Peb_G^d[\oplus]$ and not the "standard" pebbling contradictions $Peb_G^d[\vee]$ defined in terms logical or that were used in [BS02, Nor06, NH08].

Theorem 6.1. *For every clause configuration $\mathbb{C}$ that is derived from $*Peb_G^d[\oplus]$ with $d > 1$, it holds that*

$$|\mathbb{C}| > \textsf{cost}(\mathbb{R}(\mathbb{C})) \enspace ,$$

where $\textsf{cost}(\mathbb{R}(\mathbb{C})) = \left|\bigcup_{[B]\langle W\rangle\in\mathbb{R}(\mathbb{C})}(W \cup B)\right|$.

Proof. Let us write

$$V^* = \bigcup_{[B]\langle W\rangle\in\mathbb{R}(\mathbb{C})}(B \cup W) \tag{18}$$

to denote all vertices mentioned in the configuration induced by $\mathbb{C}$. At this point, we know nothing about the relationship between V^* and $V(\mathbb{C})$. However, it is intuitively plausible that $V^* \subseteq V(\mathbb{C})$, i.e., that the clause set must mention variables for the vertices on which it induces pebbles, and as we will see later in the proof this is indeed the case.

Consider the bipartite graph with clauses in $\mathbb{C}$ on the left-hand side and vertices in V^* on the right-hand side. We draw an edge between $C \in \mathbb{C}$ and $v \in V^*$ if C mentions v. That is, the set of neighbors of C is $N(C) = V(C) \cap V^*$.

Let $\mathbb{C}_1 \subseteq \mathbb{C}$ be a set of maximal size such that $|\mathbb{C}_1| > |N(\mathbb{C}_1)|$. Let $\mathbb{C}_2 = \mathbb{C} \setminus \mathbb{C}_1$ and define the vertex set $V_1^* = N(\mathbb{C}_1)$. By the maximality of $\mathbb{C}_1$ we have

$$|\mathbb{D}| \leq \left|N(\mathbb{D}) \setminus V_1^*\right| \text{ for all } \mathbb{D} \subseteq \mathbb{C}_2. \tag{19}$$

This holds trivially in the case $\mathbb{C}_2 = \emptyset$. For the case of nonempty $\mathbb{C}_2$, if, by way of contradiction, $|\mathbb{D}| > \left|N(\mathbb{D}) \setminus V_1^*\right|$, then $\mathbb{C}' = \mathbb{C}_1 \cup \mathbb{D}$ would be a larger set than $\mathbb{C}_1$ with $|\mathbb{C}'| > |N(\mathbb{C}')|$, contradicting the maximality of $\mathbb{C}_1$.

Equation (19) implies, by Hall's marriage theorem, that there is an injective mapping M of $\mathbb{C}_2$ into $V^* \setminus V_1^*$. For $C \in \mathbb{C}_2$ let $v(C) = M(C)$ be the vertex matched to C and let $V_2^* = \{v(C) \mid C \in \mathbb{C}_2\}$. We now show $V^* = V_1^* \cup V_2^*$ and this will prove the theorem because $|\mathbb{C}_1| > |V_1^*|$ and $|\mathbb{C}_2| = |V_2^*|$ imply

$$|\mathbb{C}| = |\mathbb{C}_1| + |\mathbb{C}_2| > |V_1^*| + |V_2^*| = |V^*|. \tag{20}$$

Assume by way of contradiction $V_3^* = V^* \setminus (V_1^* \cup V_2^*) \neq \emptyset$. Fix some $v \in V_3^*$ and $[B]\langle W\rangle \in \mathbb{R}(\mathbb{C})$ such that $v \in (W \cup B)$, which must exist by definition of V^*. By Definition 5.2

$$\mathbb{C} \triangleright \Big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\Big) \vee \Big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^{d} w_i\Big) \ . \tag{21}$$

We claim that we can construct a truth value assignment α that makes $\mathbb{C}$ true but $\big(\bigvee_{b \in B} \bigoplus_{i=1}^{d} b_i\big) \vee \big(\bigvee_{w \in W} \overline{\bigoplus}_{i=1}^{d} w_i\big)$ false. This clearly contradicts condition (3a) from Definition 5.2 and so the theorem follows.

The desired α will be the union of three partial assignments $\alpha_1 \cup \alpha_2 \cup \alpha_3$ that assign values to distinct variables. For $j = 1, 2$ let $B_j = B \cap V_j^*$ and $W_j = W \cap V_j^*$. By assumption $v \in (B \cup W) \setminus (B_1 \cup W_1)$ so conditions (3b), (3c) in Definition 5.2 imply

$$\mathbb{C} \nvDash \Big(\bigvee_{b \in B_1} \bigoplus_{i=1}^{d} b_i\Big) \vee \Big(\bigvee_{w \in W_1} \overline{\bigoplus}_{i=1}^{d} w_i\Big) \tag{22}$$

so we can find a truth value assignment β that sets $\mathbb{C}$ to true but violates all constraints $\bigoplus_{i=1}^{d} b_i$, $b \in B_1$, and $\overline{\bigoplus}_{i=1}^{d} w_i$, $w \in W_1$. Take α_1 to be the restriction of β to $\mathit{Vars}\big(\mathbb{C}_1\big) \cup \mathit{Vars}^d\big(B_1 \cup W_1\big)$. What is important to notice about α_1 is that it *(i)* does not assign any value to $\mathit{Vars}^d\big(V_2^* \cup V_3^*\big)$, *(ii)* sets $\mathbb{C}_1$ to true, *(iii)* violates all constraints $\bigoplus_{i=1}^{d} b_i$, $b \in B_1$, and $\overline{\bigoplus}_{i=1}^{d} w_i$, $w \in W_1$ and *(iv)* any extension of α_1 will not change *(ii)*, *(iii)*.

To construct α_2 we use the matching M of $\mathbb{C}_2$ into V_2^* to find a distinct vertex $v(C)$ for every $C \in \mathbb{C}_2$ and a literal over some variable $v(C)_i \in \mathit{Vars}^d\big(v(C)\big)$ that fixes C to true. Let γ be this partial assignment. We stress that γ assigns values to at most one variable v_i for any $v \in B_2 \cup W_2$. This means that we can extend γ to an assignment α_2 to $\mathit{Vars}^d\big(V_2^*\big)$ still satisfying $\mathbb{C}_2$ but violating all constraints $\bigoplus_{i=1}^{d} b_i$, $b \in B_2$, and $\overline{\bigoplus}_{i=1}^{d} w_i$, $w \in W_2$. Regarding α_2, notice it *(i)* assigns values only to $\mathit{Vars}^d\big(V_2^*\big)$, *(ii)* sets $\mathbb{C}_2$ to true, *(iii)* violates all constraints $\bigoplus_{i=1}^{d} b_i$, $b \in B_2$, and $\overline{\bigoplus}_{i=1}^{d} w_i$, $w \in W_2$ and *(iv)* any extension of α_2 will not change *(ii)*, *(iii)*.

Finally, to construct α_3 we pick for every $v \in (B \cup W) \cap V_3^*$ an assignment that violates the constraint over v. I.e., if $v \in B$ we set α_3 so that $\bigoplus_{i=1}^{d} v_i$ is false and if $v \in W$ set it so that $\overline{\bigoplus}_{i=1}^{d} v_i$ is false. Notice α_3 assigns values only to variables in $\mathit{Vars}^d\big(V_3^*\big)$. Thus, taking $\alpha = \alpha_1 \cup \alpha_2 \cup \alpha_3$ contradicts (21), which proves the claim. $\square$

Theorem 3.3 now follows from Theorems 5.3 and 6.1 together with Lemma 5.1.

7 From Res-Pebblings to Black-White Pebblings

To complete the proof of Theorem 1.1, we also need to establish lower bounds on res-pebbling price in terms of black-white pebbling price.

Theorem 3.4 (restated). *For any DAG G it holds that $\textsf{Res-Peb}(G) \geq \textsf{BW-Peb}(G)$.*

On the face of it, the resolution-pebbling game might seem quite different from the standard black-white pebble game. The lower bounds on black-white pebbling depend critically on the fact that the rules for black pebble placement and white pebble removal are very strict. In the resolution-pebbling game, however, we can always remove any white pebbles by doing an erasure, and by weakening we can always black-pebble any vertex although no white pebbles are even near this vertex. However, the fact that we collect black pebbles B and white pebbles W in subconfigurations $[B]\langle W\rangle$, and only allow operations on these subconfigurations, makes it relatively straightforward to show Theorem 3.4. The proof follows immediately from the following pair of lemmas, proved next.

Lemma 7.1. *Given any complete res-pebbling $\mathcal{R}$ of G using weakening, there is a complete res-pebbling $\mathcal{R}'$ which never makes any weakening moves and has $\textsf{cost}(\mathcal{R}') \leq \textsf{cost}(\mathcal{R})$.*

Lemma 7.2. *Given any complete res-pebbling $\mathcal{R}'$ of G that does not make any weakening moves, there is a complete standard black-white pebbling $\mathcal{P}$ of G such that $\textsf{cost}(\mathcal{P}) \leq \textsf{cost}(\mathcal{R}')$.*

Proof of Lemma 7.1. This is true since we can always construct a shadow pebbling that matches download, resolution, and erasure moves but ignores weakening moves. Such a pebbling can have at most the same cost as the pebbling that it is shadowing.

Formally, given any complete res-pebbling $\mathcal{R} = \{\mathbb{R}_0, \ldots, \mathbb{R}_\tau\}$ of G, we construct our pebbling $\mathcal{R}' = \{\mathbb{R}'_0, \ldots, \mathbb{R}'_\tau\}$ inductively by maintaining the following invariant: For every $\mathbb{R}_t \in \mathcal{R}$ there is a surjective function $g_t : \mathbb{R}_t \mapsto \mathbb{R}'_t$ such that whenever $g_t([B]\langle W\rangle) = [b]\langle W_b\rangle$ it holds that $b \in B$ and $W_b \subseteq W$. If we can construct such a function g_t for every t we are clearly done, since $\textsf{cost}(\mathbb{R}'_t) = \textsf{cost}(g_t(\mathbb{R}_t)) \leq \textsf{cost}(\mathbb{R}_t)$ and we must have $g_\tau([z]\langle\emptyset\rangle) = \{[z]\langle\emptyset\rangle\}$. The base case $\mathbb{R}_0 = \emptyset$ is trivial. We make a case analysis over the pebbling move made at time t.

Download $\mathbb{R}_t = \mathbb{R}_{t-1} \cup \{[v]\langle \mathit{pred}(v)\rangle\}$: Make the same download move in $\mathcal{R}'$, set $g_t([v]\langle \mathit{pred}(v)\rangle) = [v]\langle \mathit{pred}(v)\rangle$ and let $g_t = g_{t-1}$ for all other subconfigurations in $\mathbb{R}_{t-1}$.

Erasure $\mathbb{R}_t = \mathbb{R}_{t-1} \setminus \{[B]\langle W\rangle\}$: Set $\mathbb{R}'_t = g_{t-1}(\mathbb{R}_t)$ (which might result in an erasure or leave $\mathbb{R}'_t = \mathbb{R}'_{t-1}$ unchanged).

Weakening $\mathbb{R}_t = \mathbb{R}_{t-1} \cup \{[B \cup B']\langle W \cup W'\rangle\}$ for some subconfiguration $[B]\langle W\rangle \in \mathbb{R}_{t-1}$: set

$$g_t([B \cup B']\langle W \cup W'\rangle) = g_{t-1}([B]\langle W\rangle)$$

and let $g_t = g_{t-1}$ for all other subconfigurations (leaving $\mathbb{R}'_t = \mathbb{R}'_{t-1}$ unchanged).

Resolution $\mathbb{R}_t = \mathbb{R}_{t-1} \cup \{[B_1 \cup B_2]\langle W_1 \cup W_2\rangle\}$ derived from $[B_1]\langle W_1 \cup \{v\}\rangle, [B_2 \cup \{v\}]\langle W_2\rangle \in \mathbb{R}_{t-1}$: This is the only nontrivial case. Let $g_{t-1}([B_1]\langle W_1 \cup \{v\}\rangle) = [b_1]\langle W'_1\rangle$ and similarly let $g_{t-1}([B_2 \cup \{v\}]\langle W_2\rangle) = [b_2]\langle W'_2\rangle$. Note that by the induction hypothesis we have $b_1 \in B_1 \subseteq B_1 \cup B_2$ and $W'_2 \subseteq W_2 \subseteq W_1 \cup W_2$. We get three subcases:

1. $v \notin W'_1$: Then $W'_1 \subseteq W_1 \subseteq W_1 \cup W_2$, so we can set $g_t([B_1 \cup B_2]\langle W_1 \cup W_2\rangle) = [b_1]\langle W'_1\rangle$.
2. $v \neq b_2$: Then $b_2 \in B_2 \subseteq B_1 \cup B_2$, so we can set $g_t([B_1 \cup B_2]\langle W_1 \cup W_2\rangle) = [b_2]\langle W'_2\rangle$.

3. Otherwise, we have $v = b_2$ and $v \in W_1'$, so we can resolve $[b_1]\langle W_1' \rangle$ and $[b_2]\langle W_2' \rangle$ to get $[b_1]\langle (W_1' \cup W_2') \setminus \{b_2\} \rangle$ and set $g_t([B_1 \cup B_2]\langle W_1 \cup W_2 \rangle) = [b_1]\langle (W_1' \cup W_2') \setminus \{b_2\} \rangle$.

Let $g_t = g_{t-1}$ for all other subconfigurations in $\mathbb{R}_{t-1}$.

Since in all cases we can construct a surjective function $g_t : \mathbb{R}_t \mapsto \mathbb{R}_t'$ satisfying the invariant conditions, the lemma follows. □

Proof of Lemma 7.2. We assume without loss of generality that $\mathcal{R}'$ terminates at time τ once it contains a subconfiguration $[z]\langle \emptyset \rangle$ where z is the sink of G. Next, we define the *essential* subconfigurations of $\mathcal{R}'$ by backwards induction as follows. The only essential subconfiguration of $\mathbb{R}_\tau$ is $[z]\langle \emptyset \rangle$. For $t < \tau$, we say a subconfiguration is essential in $\mathbb{R}_t$ iff it is either *(i)* essential at time $t+1$, or *(ii)* one of the two subconfigurations used in a resolution step resulting in an essential subconfiguration. To prove the lemma it is sufficient to show that the set of pebbles mentioned in essential subconfigurations forms a legal black-white pebbling of G. Formally, let

$$\mathbb{B}_t = \{\cup B \mid [B]\langle W \rangle \text{ is essential in } \mathbb{R}_t\}$$

and

$$\mathbb{W}_t = \{\cup W \mid [B]\langle W \rangle \text{ is essential in } \mathbb{R}_t\} \setminus \mathbb{B}_t.$$

We claim the sequence $\{(\mathbb{B}_0, \mathbb{W}_0), \ldots, (\mathbb{B}_\tau, \mathbb{W}_\tau)\}$ is a legal black-white pebbling of G and this proves our lemma.

By construction $\mathbb{B}_0 = \mathbb{W}_0 = \emptyset$ and $\mathbb{B}_\tau = \{z\}, \mathbb{W}_\tau = \emptyset$ so we only need to argue that intermediate steps are legal black-white moves. By definition of essentiality we do not need to worry about erasure moves because only unessential clauses can be erased. Thus, if the t^{th} step is an erasure then $(\mathbb{B}_{t-1}, \mathbb{W}_{t-1}) = (\mathbb{B}_t, \mathbb{W}_t)$. By assumption, there are no weakening moves so we only need to handle downloads and resolution steps which is what we do next.

Download Suppose the t^{th} step is a download of an essential subconfiguration corresponding to vertex v. Then $\mathbb{B}_t = \mathbb{B}_{t-1} \cup \{v\}$ and $\mathbb{W}_t = (\mathbb{W}_t \cup \mathit{pred}(v)) \setminus (\mathbb{B}_{t-1} \cup) \{v\})$ and this transition corresponds to a sequence of legal pebbling moves involving *(i)* placing white pebbles on all predecessors of v that are not covered by $\mathbb{W}_t \cup \mathbb{B}_t$, *(ii)* removing a white pebble from v, if $v \in \mathbb{W}_t$, which is legal because all of v's predecessors are pebbled, and *(iii)* placing a black pebble on v. Notice the overall number of pebbles throughout this sequence is at most $|\mathbb{B}_t \cup \mathbb{W}_t|$.

Resolution Suppose the t^{th} move is a resolution step deriving an essential subconfiguration. By definition, the two subconfigurations used in the resolution step are essential at time $t-1$. Furthermore, if v is the vertex that is removed in this step we have $v \in \mathbb{B}_{t-1}$. Inspection reveals $\mathbb{B}_{t-1} \supseteq \mathbb{B}_t \supseteq \mathbb{B}_{t-1} \setminus \{v\}$ and $\mathbb{W}_t \supseteq \mathbb{W}_{t-1}$ implying we can reach $(\mathbb{B}_t, \mathbb{W}_t)$ by a legal sequence of pebbling moves because we need only remove the black pebble from v and perhaps place a white one on it. This completes the proof of the lemma and with it the proof of Theorem 3.4 is complete. □

8 Concluding Remarks

We have proven an asymptotically optimal separation of space and length in resolution. This answers an open question discussed in, for instance, [ET03, Seg07, Tor04].

It would be interesting to see if the proof technique used in this paper can be extended to yield length-space tradeoffs in the sense that there are CNF formulas that can be refuted in short length and small space, but where any short refutation must have large space.[5]

Another natural question is whether our lower bounds can be extended to stronger proof systems than resolution. One obvious candidate would be the k-DNF resolution proof systems $\mathfrak{R}(k)$ introduced by Krajíček [Kra01], where the lines in the proofs are k-DNF formulas instead of clauses and one can "resolve" over up to k variables simultaneously. We believe that XOR-pebbling contradictions $Peb_G^{k+1}[\oplus]$ should separate k-DNF resolution and $(k+1)$-DNF resolution with respect to space. If so, this would establish that the k-DNF resolution proof systems form a strict hierarchy with respect to space. Currently, all that is known is the separation result in [EGM04] for the restricted case of tree-like k-DNF resolution.

Acknowledgements

The second author wants to acknowledge that Albert Atserias mentioned already back in 2006, albeit in a slightly different context, that working on XOR-pebbling contradictions might be a possible way forward.

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. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, pages 340–346, 2003.

[AD03] Albert Atserias and Victor 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. Journal version to appear in *Journal of Computer and System Sciences*.

[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.

[BKS03] Paul Beame, Henry Kautz, and Ashish Sabharwal. Understanding the power of clause learning. In *Proceedings of the 18th International Joint Conference in Artificial Intelligence (IJCAI '03)*, pages 94–99, 2003.

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

[5] At the time of submission of this paper, the answer seems to be a firm yes, but the manuscript is still in a very early stage of preparation.

[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.

[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.

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

[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.

[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.

[Gal77] Zvi Galil. On resolution with clauses of bounded size. *SIAM Journal on Computing*, 6(3):444–459, 1977.

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

[GT78] John R. Gilbert and Robert Endre Tarjan. Variations of a pebble game on graphs. Technical Report STAN-CS-78-661, Stanford University, 1978.

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

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

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

[KS07] Henry Kautz and Bart Selman. The state of SAT. *Discrete Applied Mathematics*, 155(12):1514–1524, June 2007.

[NH08] Jakob Nordström and Johan Håstad. Towards an optimal separation of space and length in resolution. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*. ACM, 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.

[Nor08] Jakob Nordström. *Short Proofs May Be Spacious: Understanding Space in Resolution*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, May 2008. Available at `http://www.csc.kth.se/~jakobn/research/`.

[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.

[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.

[Sab05] Ashish Sabharwal. *Algorithmic Applications of Propositional Proof Complexity*. PhD thesis, University of Washington, Seattle, 2005.

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

[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.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/