Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 66 (2005)

Narrow Proofs May Be Spacious: Separating Space and Width in Resolution

Jakob Nordström

Royal Institute of Technology (KTH)
SE-100 44 Stockholm, Sweden
jakobn@kth.se

November 11, 2005

Abstract

The width of a resolution proof is the maximal number of literals in any clause of the proof. The space of a proof is the maximal number of memory cells used if the proof is only allowed to resolve on clauses kept in memory. Both of these measures have previously been studied and related to the refutation size of unsatisfiable CNF formulas. Also, the resolution refutation space of a formula has been proven to be at least as large as the refutation width, but it has remained unknown whether space can be separated from width or the two measures coincide asymptotically. We prove that there is a family of k-CNF formulas for which the refutation width in resolution is constant but the refutation space is non-constant, thus solving an open problem mentioned in several previous papers.

1 Introduction

A proof system for a language L is a polynomial-time algorithm V such that for all $x \in L$ there is a string π (a *proof*) for which $V(x, \pi) = 1$. For $x \notin L$, it should hold for all strings π that $V(x, \pi) = 0$. The complexity of a proof system V is the smallest bounding function $g : \mathbb{N} \mapsto \mathbb{N}$ such that $x \in L$ if and only if there is a proof π of size $|\pi| \leq g\bigl(|x|\bigr)$ for which $V\bigl(x, \pi\bigr) = 1$. If a proof system is of polynomial complexity, it is said to be polynomially bounded. A *propositional proof system* is a proof system for tautologies in propositional logic.

The central task of proof complexity is to construct and investigate the power of different propositional proof systems. This is done for at least two reasons.

The first reason is the connection to the question of P versus NP, which is recognized as a major open problem in computational complexity theory and mathematics. Since NP is exactly the set of languages with polynomially bounded proof systems, and since TAUTOLOGY can be seen to be the dual problem of SATISFIABILITY, we have the famous theorem of Cook and Reckhow [19] that $\mathsf{NP} = \mathsf{co\text{-}NP}$ if and only if there exists a polynomially bounded propositional proof system. Thus, if it could be shown that there are no polynomially bounded proof systems for propositional tautologies, $\mathsf{P} \neq \mathsf{NP}$ would follow as

ISSN 1433-8092

a corollary since P is closed under complement. One way of approaching this distant goal is to study stronger and stronger proof systems and try to prove superpolynomial lower bounds on proof size. However, despite the fact that the last decade has seen some impressive successes for a variety of propositional proof systems, it seems that we are still very far from fully understanding the reasoning power of even quite simple ones.

The second reason is that designing efficient algorithms for proving tautologies is a very important problem not only in theoretical computer science but also in applied research and in industry, for instance in the context of formal methods. All automated theorem provers, regardless of whether they actually produce a written proof or not, explicitly or implicitly define a system in which proofs are searched for and rules which determine what proofs in this system look like. Lower bounds on proofs in such proof systems give lower bounds on the running time of corresponding automated theorem provers. In the other direction, theoretical upper bounds on proof size in a system can give upper bounds on the running time of a proof search algorithm, provided that the algorithm can be shown to search for proofs in the system in an efficient manner.

Also, the field of proof complexity has rich connections to cryptography, artificial intelligence and mathematical logic. Some good surveys of proof complexity are [6, 8, 17, 42].

Any propositional logic formula can be converted to a formula in conjunctive normal form that is only linearly larger and is unsatisfiable if and only if the original formula is a tautology. Therefore, any sound and complete system which produces refutations of unsatisfiable formulas in conjunctive normal form can be considered as a general propositional proof system.

One such proof system, which is the focus of this paper, is *resolution*. The resolution proof system appeared in [13], and began to be studied in connection with automated theorem proving in [21, 22, 37]. Because of the simplicity of resolution—there is only one derivation rule—and because all lines in a proof are clauses, this system is well adapted to proof search algorithms. Many real-world automated theorem provers are based on resolution.

Being so simple and fundamental, resolution was a natural target to attack when trying to prove lower bounds in proof complexity. In this context, it is most straightforward to prove bounds on the *length* of proofs, i.e., the number of clauses, which is easily seen to be polynomially related to the proof size. In 1968, Tseitin [40] presented a superpolynomial lower bound on refutation length in resolution, but it was not until almost 20 years later that Haken [28] proved the first exponential lower bound, which has later been followed by many similar results, for instance in [7, 12, 18, 36, 41].

A second complexity measure for resolution refutations other than length is the minimal *width*, measured as the maximal size of a clause in the refutation. This measure was first made explicit by Galil [26]. Ben-Sasson and Wigderson [12] showed that it was strongly correlated to proof length by proving that the width $W(F \vdash 0)$ of refuting a k-CNF formula F over n variables is bounded by the refutation length $L(F \vdash 0)$ by $W(F \vdash 0) = \mathrm{O}\big(\sqrt{n \log L(F \vdash 0)}\big)$, thus providing a new method for proving lower bounds on proof length by proving lower bounds on width.

The results on width lead to the question of whether other complexity measures could yield interesting insights as well. In [24, 38], Esteban and Torán introduced the concept of *space* in resolution, transforming a previous definition from [30]. Intuitively, the space of a resolution proof is the maximal number of clauses one needs to keep in memory while verifying the proof. A number of upper and lower bounds for proof space in resolution and other proof systems were subsequently presented in for instance [2, 10, 23, 25]. In several of these papers it was noted that the lower bounds on resolution refutation space for different formula families matched known lower bounds on refutation width. Atserias and Dalmau [4] showed that this was not a coincidence, but that the minimal refutation space $Sp\left(F \vdash 0\right)$ of any unsatisfiable k-CNF formula F is at least as large as the minimal refutation width $W\left(F \vdash 0\right)$ minus a constant.

An immediate follow-up question to this is whether the lower bound on space in terms of width is asymptotically strict. That is, does there exist a family $\left\{F_n\right\}_{n=1}^{\infty}$ of k-CNF formulas such that $Sp\left(F_n \vdash 0\right) = \omega\left(W\left(F_n \vdash 0\right)\right)$ or does it always hold that $Sp\left(F_n \vdash 0\right) = \mathrm{O}\left(W\left(F_n \vdash 0\right)\right)$?

Another natural question concerns the relation between space and length. It is not too hard to see that upper bounds on width imply upper bounds on length, and as a consequence of the result in [4] this must be true for space with respect to length as well. In the other direction, we have the result from [12] stated above that upper bounds on length imply upper bounds on width. Is there a similar Ben-Sasson-Wigderson-style upper bound on space in terms of length, or can short resolution proofs be arbitrarily complex with respect to space?

A third, intimately connected question is determining the refutation clause space of *pebbling contradictions* defined in terms of pebble games on directed acyclic graphs. Non-constant lower bounds on the space of refuting pebbling contradictions would separate space and width, and possibly also clarify the relation between space and length if the bounds were good enough. On the other hand, a constant upper bound on the refutation space would improve the trade-off results for different measures in resolution in [9].

The above three questions have been mentioned as interesting open problems in [9, 23, 25, 39].

In this paper, we answer the first question by separating space and width. This is done by proving an asymptotically tight bound on space for pebbling contradictions over binary trees, thus at least partially solving the open problem about the space complexity of pebbling contradictions as well. More precisely, our results are as follows (formal definitions are given in Sections 2 and 5).

Theorem 1.1. *Let T_h denote the complete binary tree of height h and $Peb^d_{T_h}$ the pebbling contradiction of degree $d \geq 2$ defined over T_h. Then the space of refuting $Peb^d_{T_h}$ by resolution is $Sp\left(Peb^d_{T_h} \vdash 0\right) = \Theta(h)$.*

Corollary 1.2. *For all $k \geq 4$, there is a family $\left\{F_n\right\}_{n=1}^{\infty}$ of k-CNF formulas of size $\mathrm{O}(n)$ such that $W(F_n \vdash 0) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Theta(\log n)$.*

The organization of this paper is as follows. We start by presenting the resolution proof system in Section 2. In Section 3, we gather some structural results for CNF formulas which might be of independent interest. Section 4 gives a short introduction to pebble games, and in Section 5 we review some previous results connecting resolution and pebbling. The bound on refutation space which separates space and width is then proven in three steps.

- First, we define a modified pebble game and establish a lower bound for this game in terms of the standard black-white pebble game (Sections 6 and 7).
- Next, we show that a resolution refutation of a pebbling contradiction induces a pebbling of the underlying graph in our modified pebble game (Section 8).
- Finally, we prove that if a set of clauses induces many pebbles, the set must contain at least as many clauses. Since a resolution refutation induces a pebbling, and such a pebbling must contain many pebbles at some point, we deduce that the clause space of the resolution derivation must be large (Section 9).

We conclude in Section 10 by giving suggestions for further research.

2 The Resolution Proof System

A *literal* is either a propositional logic variable or its negation, denoted x and $\overline{x}$ respectively (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 $a \neq \overline{b}$.

A *clause* $C = a_1 \vee \ldots \vee a_k$ is a set of literals. 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* $F = C_1 \wedge \ldots \wedge C_m$ is a set of clauses. A *k-CNF formula* is a CNF formula consisting of k-clauses.

In the following, we let 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 define

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

We let $Vars(C)$ denote the set of variables and $Lit(C)$ the set of literals in a clause C. (Although the notation $Lit(C)$ is slightly redundant given the definition of a clause as a set of literals, we include it for clarity.) This notation is extended to sets of clauses by taking unions.

A *resolution derivation* $\pi : F \to A$ 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, $1 \leq i \leq \tau$, is either 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*

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

We refer to (1) as *resolution on the variable* x and $B \vee C$ 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 (the clause with no literals) from F.

For F a formula and $\mathcal{G} = \{G_1, \ldots, G_n\}$ a set of formulas, 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.

Resolution is sound and implicationally complete. That is, if there is a resolution derivation $\pi : F \to A$ then $F \vDash A$, and if $F \vDash A$ then there is a resolution derivation $\pi : F \to A'$ for some $A' \subseteq A$. In particular, F is unsatisfiable if and only if there is a resolution refutation of F.

We can associate with every resolution derivation $\pi : F \to A$ a directed acyclic graph (DAG) G_π, with the clauses in π labelling the vertices and with edges from the assumption clauses to the resolvent for each application of the resolution rule. 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* $L(F)$ of a CNF formula F is the number of clauses in it, and for π a resolution derivation $L(\pi)$ is the number of clauses in π. The length of deriving a clause A from a formula F is $L(F \vdash A) = \min_{\pi : F \to A} \{L(\pi)\}$, where the minimum is taken over all resolution derivations of A, and the length of refuting F by resolution is $L(F \vdash 0)$. The 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 $|C|$. 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 $W(F \vdash A) = \min_{\pi : F \to A} \{ W(\pi) \}$, and the width of refuting F is $W(F \vdash 0)$.

If a resolution refutation has constant width, it must be of size polynomial in the number of variables. Conversely, if all refutations of a formula are very wide, it seems reasonable that any refutation of this formula must be very long as well. This intuition is made precise in the following theorem.

Theorem 2.1 ([12]). *The width of refuting a CNF formula F is bounded from above by*

$$W(F \vdash 0) \leq W(F) + \mathrm{O}\left(\sqrt{n \log L(F \vdash 0)}\right),$$

where n is the number of variables in F.

In [15], it was shown that this bound on width in terms of length is essentially optimal.

We next define the measure of *space*. Following the exposition in [24], 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 maximal number of clauses that need to be kept in memory simultaneously during a verification of the proof. The variable space is the maximal "total" space needed, where also the width of the clauses is taken into account.

For the formal definition, it is convenient to use the following alternative definition of resolution introduced by [2]. We employ the standard notation $[n] = \{1, 2, \ldots, n\}$.

Definition 2.2 (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 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 $D \notin \mathbb{C}_{t-1}$ inferred by resolution from $C_1, C_2 \in \mathbb{C}_{t-1}$.

A resolution derivation $\pi : F \to A$ 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 0 from F.

Definition 2.3 (Clause space [2, 9]). The *clause space* of a resolution derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is $\max_{t \in [\tau]}\{|\mathbb{C}_t|\}$. The clause space of deriving A from F is $Sp(F \vdash A) = \min_{\pi : F \to A}\{Sp(\pi)\}$. $Sp(F \vdash 0)$ is the minimal clause space of any resolution refutation of F.

Definition 2.4 (Variable space [2]). The *variable space* of a configuration $\mathbb{C}$ is $VarSp(\mathbb{C}) = \sum_{C \in \mathbb{C}} W(C)$. The variable space of a resolution derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is $\max_{0 \le i \le s}\{VarSp(\mathbb{C}_i)\}$, and $VarSp(F \vdash 0)$ is the minimal variable space of any resolution refutation of F.

Restricting the resolution derivations to tree-like resolution, we get the measures $Sp_{\mathfrak{T}}(F \vdash 0)$ and $VarSp_{\mathfrak{T}}(F \vdash 0)$ in analogy with $L_{\mathfrak{T}}(F \vdash 0)$ defined above.

All contradictory CNF formulas can be refuted in clause space linear in the formula size. More precisely:

Theorem 2.5 ([24]). *Any unsatisfiable CNF formula F on n variables can be refuted in clause space $n + 2$.*

Theorem 2.6 ([24]). *Any unsatisfiable CNF formula F with m clauses can be refuted in clause space $m + 1$, i.e., $Sp(F \vdash 0) \le L(F) + 1$.*

Thus the interesting question is which formulas demand this much space, and which formulas can be refuted in for instance logarithmic or even constant space. It has been shown that there are polynomial-size formulas that meet the upper bounds of Theorems 2.5 and 2.6 up to a multiplicative constant.

Theorem 2.7 ([2, 38]). *There is a polynomial-size family $\{F_n\}_{n=1}^{\infty}$ of unsatisfiable 3-CNF formulas such that $Sp(F \vdash 0) = \Omega\big(L(F)\big) = \Omega\big(|\,Vars(F)|\big)$.*

Lower bounds on clause space have been presented for a number of different CNF formula families [2, 10, 38]. As was mentioned above, in these papers it was observed that the lower bounds on refutation space coincided with the lower bounds on refutation width. This lead to the conjecture that the width measure is a lower bound for the clause space measure, a conjecture that was proven true in [4].

Theorem 2.8 ([4]). *Let F be an arbitrary unsatisfiable CNF formula. Then it holds that $Sp(F \vdash 0) - 3 \ge W(F \vdash 0) - W(F)$.*[1]

In other words, the extra clause space exceeding the minimum 3 needed for any resolution derivation is bounded from below by the extra width exceeding the width of the formula. An immediate corollary of this theorem is that for

[1]The statement of the theorem in [4] is $Sp(F \vdash 0) \ge W(F \vdash 0) - W(F)$, but this can be sharpened by a constant if one does the calculations carefully.

polynomial-size k-CNF formulas, constant clause space implies polynomial proof length.

A very natural question, which has remained open, is what holds in the other direction. Do the space and width measures coincide asymptotically or is there a formula family separating space from width? We remark that in order for this question to be interesting, we should restrict our attention to families of k-CNF formulas. Any resolution refutation of an unsatisfiable CNF formula F with *minimum* clause width k can be shown to require clause space at least $k+2$ (see [24]), so it is easy to find CNF formulas $\{F_n\}_{n=1}^{\infty}$ of growing width such that $W(F_n \vdash 0) - W(F_n) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Omega(n)$.

In this paper, we settle the open question of the relationship between space and width by proving that there is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ such that $W(F_n \vdash 0) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Theta(\log L(F_n))$.

3 On the Structure of Unsatisfiable CNF Formulas

In this section, we show some easy technical results about CNF formulas that will be needed later on in the paper. We also prove two theorems about the structure of unsatisfiable CNF formulas that might be interesting in their own right (Theorems 3.6 and 3.10).

Definition 3.1. A *restriction* ρ is a partial truth value assignment. We represent a restriction as the set of literals $\rho = \{a_1, \ldots, a_m\}$ set to true by ρ. For a clause C, the *ρ-restriction* of C is

$$C|_{\rho} = \begin{cases} 1 & \text{if } \rho \cap \mathit{Lit}(C) \neq \emptyset, \\ C \setminus \{\overline{a} \mid a \in \rho\} & \text{otherwise,} \end{cases}$$

where 1 denotes the trivially true clause, and the ρ-restriction $\mathbb{C}|_{\rho}$ of a set of clauses $\mathbb{C}$ is the union of the ρ-restrictions $C|_{\rho} \neq 1$ for $C \in \mathbb{C}$.

We write $\rho(\neg C) = \{\overline{a} \mid a \in \mathit{Lit}(C)\}$ to denote the unique minimal restriction that falsifies C.

Definition 3.2. We say that a set of clauses $\mathbb{C}$ implies a clause D *minimally* if $\mathbb{C} \vDash D$ but for all $\mathbb{C}' \subsetneqq \mathbb{C}$ it holds that $\mathbb{C}' \nvDash D$. If $\mathbb{C} \vDash 0$ minimally, $\mathbb{C}$ is said to be *minimally unsatisfiable*.

Lemma 3.3. *Suppose for $\mathbb{C}$ a set of clauses and D a clause that $\mathbb{C} \vDash D$ minimally, and let $\rho = \rho(\neg D)$. Then $\mathbb{C}|_{\rho}$ is minimally unsatisfiable. Also, it holds that $\left|\mathbb{C}|_{\rho}\right| = \left|\mathbb{C}\right|$, i.e., no literal $a \in \mathit{Lit}(D)$ occurs negated in $\mathbb{C}$.*

Proof. If $\mathbb{C} \vDash D$, then clearly $\mathbb{C}|_{\rho} \vDash D|_{\rho}$ for any restriction ρ. In particular, for $\rho = \rho(\neg D)$ the set of clauses $\mathbb{C}|_{\rho}$ is unsatisfiable. Pick any $\mathbb{C}' \subseteq \mathbb{C}$ such that $\mathbb{C}'|_{\rho}$ is minimally unsatisfiable. If there was a truth value assignment α such that $\alpha\bigl(\mathbb{C}'\bigr) = 1$ and $\alpha\bigl(D\bigr) = 0$, this α would satisfy $\mathbb{C}'|_{\rho}$, which is contrary to assumption. Hence $\mathbb{C}' \vDash D$, and again by assumption we must have $\mathbb{C}' = \mathbb{C}$. This also shows that $\left|\mathbb{C}|_{\rho}\right| = \left|\mathbb{C}\right|$, for if ρ satisfied some clause $C \in \mathbb{C}$ this would imply that $(\mathbb{C} \setminus \{C\})|_{\rho}$ was minimally unsatisfiable for $\mathbb{C} \setminus \{C\} = \mathbb{C}' \subsetneqq \mathbb{C}$. ☐

Lemma 3.4. *Suppose for a set of clauses $\mathbb{C}$ and clauses D_1 and D_2 with $\mathit{Vars}(D_1) \cap \mathit{Vars}(D_2) = \emptyset$ that $\mathbb{C} \vDash D_1 \vee D_2$ but $\mathbb{C} \nvDash D_2$. Then there is a literal $a \in \mathit{Lit}(\mathbb{C}) \cap \mathit{Lit}(D_1)$.*

Proof. Pick a truth value assignment α such that $\alpha\big(\mathbb{C}\big) = 1$ but $\alpha\big(D_2\big) = 0$. By assumption $\alpha\big(D_1\big) = 1$. Let α' be the same assignment except that all satisfied literals in D_1 are flipped to false. Then $\alpha'\big(D_1 \vee D_2\big) = 0$ forces $\alpha'\big(\mathbb{C}\big) = 0$, so the flip must have falsified some previously satisfied clause in $\mathbb{C}$. ☐

Lemma 3.5. *Let $\mathbb{C}$ be a set of clauses and D a clause such that $\mathbb{C} \vDash D$ minimally and $a \in Lit(\mathbb{C})$ but $\overline{a} \notin Lit(\mathbb{C})$. Then $a \in Lit(D)$.*

Proof. Suppose not. Let $\mathbb{C}_1 = \big\{C \in \mathbb{C} \mid a \in Lit(C)\big\}$ and $\mathbb{C}_2 = \mathbb{C} \setminus \mathbb{C}_1$. Since $\mathbb{C}_2 \nvDash D$ there is an α such that $\alpha(\mathbb{C}_2) = 1$ and $\alpha(D) = 0$. Note that $\alpha(a) = 0$, since otherwise $\alpha(\mathbb{C}_1) = 1$. It follows that $\overline{a} \notin Lit(D)$. Flip a to true. By construction $\alpha^{a=1}\big(\mathbb{C}_1\big) = 1$, and $\mathbb{C}_2$ and D are not affected since $\{a, \overline{a}\} \cap \big(Lit(\mathbb{C}_2) \cup Lit(D)\big) = \emptyset$, so $\alpha^{a=1}\big(\mathbb{C}\big) = 1$ and $\alpha^{a=1}\big(D\big) = 0$. Contradiction. ☐

The fact that a minimally unsatisfiable CNF formula F must have more clauses than variables seems to have been proven independently a number of times (see e.g. [1, 5, 32]). We extend this result to subsets of variables in a minimally implicating CNF formula and the clauses containing variables from these subsets.

Theorem 3.6. *Suppose that F is CNF formula that implies a clause D minimally. For V any subset of variables, let $F_V = \{C \in F \mid Vars(C) \cap V \neq \emptyset\}$. Then if $V \subseteq Vars(F) \setminus Vars(D)$, it holds that $|F_V| > |V|$. In particular, if F is minimally unsatisfiable we have $|F_V| > |V|$ for all $V \subseteq Vars(F)$.*

Proof. By induction over $V \subseteq Vars(F) \setminus Vars(D)$.

If $|V| = 1$, then $|F_V| \geq 2$, since any $x \in V$ must occur both positively and negatively in F by Lemma 3.5.

The inductive step just generalizes the proof of this lemma. Suppose that $|F_{V'}| > |V'|$ for all strict subsets $V' \subsetneqq V \subseteq Vars(F) \setminus Vars(D)$ and consider V. Since $F_{V'} \subseteq F_V$ if $V' \subseteq V$, choosing any V' of size $|V| - 1$ we see that $|F_V| \geq |F_{V'}| \geq |V'| + 1 = |V|$.

If $|F_V| > |V|$ there is nothing to prove, so assume that $|F_V| = |V|$. Consider the bipartite graph with the variables V and the clauses in F_V as vertices, and edges between variables and clauses for all variable occurrences. Since for all $V' \subseteq V$ the set of neighbours $N(V') = F_{V'} \subseteq F_V$ satisfies $|N(V')| \geq |V'|$, by Hall's Marriage Theorem there is a perfect matching between V and F_V. Use this matching to satisfy F_V assigning values to variables in V only.

The clauses in $F' = F \setminus F_V$ are not affected by this partial truth value assignment, since they do not contain any occurrences of variables in V. Furthermore, by the minimality of F it must hold that F' can be satisfied and D falsified simultaneously by assigning values to variables in $Vars(F') \setminus V$.

The two partial truth value assignments above can be combined to an assignment that satisfies all of F but falsifies D, which is a contradiction. Thus $|F_V| > |V|$. The theorem follows. ☐

It is easy to see that Theorem 3.6 does not hold if we drop the condition that F should imply D minimally. However, if D is derivable from F by a resolution derivation that uses all clauses in F, we can obtain a completely analogous bound. We conclude the section by proving this result, which is an extension of a theorem in [5].

For a resolution derivation $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ of D from F in the sense of Definition 2.2, the derivation graph G_π is uniquely determined. The clause $C \in \mathbb{C}_t$ is said to be *superfluous* if there is no path in G_π from the vertex for $C \in \mathbb{C}_t$ to the vertex for D. A derivation is *non-superfluous* if it contains no superfluous clauses. For every derivation $\pi : F \to D$ we can get a corresponding non-superfluous derivation $\pi' : F \to D$ in at most the same length, width and space by only considering the vertices in G_π from which D is reachable. In the following, we therefore assume without loss of generality that all derivations are non-superfluous.

Definition 3.7. The clause $C \in F$ is *used* in a resolution derivation π if C occurs in π. The derivation $\pi : F \to D$ *uses exactly* $F' \subseteq F$ if π uses all clauses $C' \in F'$ but no clauses $C \in F \setminus F'$.

We say that a CNF formula F *syntactically precisely yields* the clause D if there is a non-superfluous resolution derivation $\pi : F \to D$ that uses exactly F, and denote this by $F \vDash_{\triangledown} D$. If $F \vDash_{\triangledown} 0$, we say that F is *syntactically precisely contradictory*.

The intuition behind Theorem 3.6 is that if a set of clauses is strictly smaller than the number of distinct variables in the set, there simply are not enough clauses to eliminate all variables from the right-hand side of the implication. The intuition is the same when we move over to resolution derivations, but here we have to express this fact syntactically.

Definition 3.8. A set of *strictly distinct literal representatives* for a set of CNF clauses $\mathbb{C} = \{C_1, \ldots, C_m\}$ is a set of literals $\{x_1^{\nu_1}, \ldots, x_m^{\nu_m}\}$ such that $x_i^{\nu_i} \in \mathit{Lit}(C_i)$ and $x_i \neq x_j$ if $i \neq j$.

Lemma 3.9. *Suppose that $\mathbb{C} = \{C_1, \ldots, C_m\}$ is a set of CNF clauses with strictly distinct literal representatives $\{x_1^{\nu_1}, \ldots, x_m^{\nu_m}\}$ and that $\mathbb{D}$ is another set of CNF clauses such that $\mathit{Vars}(\mathbb{D}) \cap \{x_i \mid i \in [m]\} = \emptyset$. Then for every resolution derivation $\pi : \mathbb{C} \cup \mathbb{D} \to A$ that uses some clause from $\mathbb{C}$, the clause A must contain at least one of the literal representatives $x_i^{\nu_i}$.*

Proof. By induction over π. Case analysis:

1. If a clause D is derived from $\mathbb{D}$ only, clearly it cannot contain any representative variable x_i.

2. If B is derived by resolution from C and D where C is obtained from $\mathbb{C} \cup \mathbb{D}$ and D is obtained from $\mathbb{D}$ only, by induction C contains a literal representative $x_i^{\nu_i}$ and $x_i \notin \mathit{Vars}(D)$. Hence the literal $x_i^{\nu_i}$ is not eliminated in the resolution step.

3. If B is derived by resolution from C and D where both clauses have been obtained from $\mathbb{C} \cup \mathbb{D}$, by induction there are literal representatives $x_i^{\nu_i} \in \mathit{Lit}(C)$ and $x_j^{\nu_j} \in \mathit{Lit}(D)$. We know by assumption that $x_i^{\nu_i}$ and $x_j^{\nu_j}$ are not complementary, i.e., $x_i^{\nu_i} \neq x_j^{1-\nu_j}$. If the variable resolved over is one of x_i and x_j, say x_i, then the other literal $x_j^{\nu_j}$ stays in B. Otherwise, both $x_i^{\nu_i}$ and $x_j^{\nu_j}$ are present in B.

Since by assumption π uses some clause from $\mathbb{C}$, the final clause A must contain some literal representative. The lemma follows. ∎

Theorem 3.10. *Suppose for F a CNF formula and D a clause that $F \vDash_{\triangledown} D$, and let $F_V = \{C \in F \mid \mathit{Vars}(C) \cap V \neq \emptyset\}$. Then if $V \subseteq \mathit{Vars}(F) \setminus \mathit{Vars}(D)$, it holds that $|F_V| > |V|$. In particular, if $F \vDash_{\triangledown} 0$ then $|F_V| > |V|$ for all $V \subseteq \mathit{Vars}(F)$.*

Proof. The proof is by pattern matching on the proof of Theorem 3.6.

Suppose by induction that $|F_{V'}| > |V'|$ for all non-empty strict subsets $V' \subsetneqq V \subseteq \mathit{Vars}(F) \setminus \mathit{Vars}(D)$ and consider V. As before, we observe that $|F_V| \geq |V|$.

If $|F_V| > |V|$ we are done, so assume that $|F_V| = |V| = m$. Considering the bipartite graph with the variables V and the clauses in F_V as vertices and edges between variables and clauses for all variable occurrences, we again get a perfect matching. This perfect matching yields a set of distinct literal representatives $\{x_i^{\nu_i} \mid i \in [m]\}$ for F_V. By construction, $\mathit{Vars}(F \setminus F_V) \cap \{x_i \mid i \in [m]\} = \emptyset$. Appealing to Lemma 3.9, we see that D must contain some variable $x_i \in V \subseteq \mathit{Vars}(F) \setminus \mathit{Vars}(D)$. Contradiction. Consequently, $|F_V| > |V|$ and the theorem follows by induction. $\square$

Of course, Theorem 3.6 above follows as a corollary of Theorem 3.10, but we felt that the simpler direct proof of the former theorem was also of independent interest.

4 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 [24]) and to construct CNF formulas which are hard for different variants of resolution in various respects (see for example [3, 11, 14, 16]). An excellent survey of pebbling up to 1980 is [35].

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 vertices with outdegree 0 *targets*.

Definition 4.1 (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 1-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- and white-pebbled vertices. 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.

A *legal black-white pebbling reaching* (B, W) in G is a sequence of configurations $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = (\emptyset, \emptyset)$, $\mathbb{P}_\tau = (B, W)$, and for all $t \in [\tau]$, $\mathbb{P}_t$ follows from $\mathbb{P}_{t-1}$ by one of the rules above.

The *cost* of a pebbling configuration $\mathbb{P} = (B, W)$ is $\mathsf{cost}(\mathbb{P}) = |B \cup W|$ and the cost of a of a legal pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is $\max_{t\in[\tau]}\{\mathsf{cost}(\mathbb{P}_t)\}$. The *black-white pebbling price* of (B, W), denoted $\mathsf{BW\text{-}Peb}(B, W)$, is the minimal cost of any legal pebbling reaching (B, W).

A *legal black-white pebbling of* G is a pebbling reaching $(\{z\}, \emptyset)$, and the *black-white pebbling price* of G, denoted $\mathsf{BW\text{-}Peb}(G)$, is the minimal cost of any legal pebbling of G.

A *legal black pebbling of* G is a pebbling reaching $(\{z\}, \emptyset)$ using black pebbles only, i.e., $W_t = \emptyset$ for all t, and the *(black) pebbling price* of G, denoted $\mathsf{Peb}(G)$, is the minimal cost of any legal black pebbling of G.

A black-white pebbling *visiting* z is a pebbling such that $\mathbb{P}_0 = \mathbb{P}_\tau = (\emptyset, \emptyset)$ and there exists a $t \in [\tau]$ such that $z \in B_t \cup W_t$. The minimum cost of such a pebbling is denoted $\mathsf{BW\text{-}Peb}^{\emptyset}(G)$.

It is easy to see that $\mathsf{BW\text{-}Peb}^{\emptyset}(G) \leq \mathsf{BW\text{-}Peb}(G) \leq \mathsf{BW\text{-}Peb}^{\emptyset}(G) + 1$.

We think of the moves in a pebbling as occurring at integral time intervals $t = 1, 2, \ldots$ and talk about the pebbling move "at time t" (which is the move resulting in configuration $\mathbb{P}_t$) or the moves "during the time interval $[t_1, t_2]$".

In this paper we will consider pebblings of complete binary trees. We let T denote a complete binary tree considered as a DAG with edges directed towards the root. We write T_h when we want to specify that the height of the tree is h. We use z to denote the unique target vertex of T, i.e., the root.

The black pebbling price of T_h can be established by induction over the tree height. We omit the easy proof.

Theorem 4.2. $\mathsf{Peb}(T_h) = h + 2$.

General bounds for the black-white pebbling price of trees of any arity were presented in [34]. We give a simplified proof with tight bounds for the case of complete binary trees.

Theorem 4.3. $\mathsf{BW\text{-}Peb}(T_h) = \lfloor \frac{h}{2} \rfloor + 3$ *and* $\mathsf{BW\text{-}Peb}^{\emptyset}(T_h) = \lfloor \frac{h-1}{2} \rfloor + 3$.

The proof is facilitated by the following proposition, which is an immediate consequence of Definition 4.1.

Proposition 4.4 ([20]). *If $\mathcal{P}$ is a black-white pebbling of a DAG G visiting the target, then one can get a dual pebbling $\overline{\mathcal{P}}$ of G by reversing the sequence of moves and switching the colours of the pebbles.*

Proof of Theorem 4.3. In all of this proof, we let z_1, z_2 denote the immediate predecessors of the root z of the tree.

We first show that $\mathsf{BW\text{-}Peb}^{\emptyset}(T_{h+2}) \geq \mathsf{BW\text{-}Peb}^{\emptyset}(T_h) + 1$. Suppose not, and let $\mathcal{P}$ be a pebbling in cost $K = \mathsf{BW\text{-}Peb}^{\emptyset}(T_h)$ for T_{h+2} making the minimum number of pebbling moves. Let T_h^i, $i \in [4]$, be the four disjoint subtrees of height h in T_{h+2}. It is easy to see that $\mathcal{P}$ restricted to $V(T_h^i)$ yields a legal pebbling of T_h^i visiting its root. It follows that there must exist distinct times t_i, $i \in [4]$, when T_h^i contains K pebbles and the rest of T_{h+2} is empty. Number the subtrees so that $t_1 < t_2 < t_3 < t_4$.

Suppose that the root z of T_{h+2} has been pebbled before time t_3. Then we can get a shorter pebbling of T_{h+2} by completing the subpebbling of T_h^3 but ignoring pebbles moves outside T_h^3 after time t_3.

Consequently, z must be pebbled for the first time after t_3. But at time t_3 the rest of the tree is empty, so in that case we can get a shorter legal pebbling by ignoring all moves outside T_h^3 before time t_3 and performing all moves in $\mathcal{P}$ after time t_3. Contradiction. Thus $\mathsf{BW\text{-}Peb}^{\emptyset}(T_{h+2}) \geq \mathsf{BW\text{-}Peb}^{\emptyset}(T_h) + 1$.

Next, it is easy to see that $\mathsf{BW\text{-}Peb}^{\emptyset}(T_{h+1}) \leq \mathsf{BW\text{-}Peb}(T_h)$. First black-pebble z_1 with using a pebbling $\mathcal{P}$ in cost $\mathsf{BW\text{-}Peb}(T_h)$. Place white pebbles on z and z_2, and then remove the pebbles from z_1 and z. Finally, use the dual pebbling $\overline{\mathcal{P}}$ to get the white pebble off z_2 in the same cost $\mathsf{BW\text{-}Peb}(T_h)$.

Since $\mathsf{BW\text{-}Peb}(T_1) = \mathsf{BW\text{-}Peb}^{\emptyset}(T_1) = 3$, it follows that $\mathsf{BW\text{-}Peb}^{\emptyset}(T_h) \geq \left\lfloor \frac{h-1}{2} \right\rfloor + 3$ and $\mathsf{BW\text{-}Peb}(T_h) \geq \left\lfloor \frac{(h+1)-1}{2} \right\rfloor + 3 = \left\lfloor \frac{h}{2} \right\rfloor + 3$. It remains to demonstrate that there are pebblings meeting these lower bounds. We construct such pebblings inductively.

Suppose for h odd that $\mathsf{BW\text{-}Peb}(T_h) = \mathsf{BW\text{-}Peb}^{\emptyset}(T_h) = \left\lfloor \frac{h-1}{2} \right\rfloor + 3 = \left\lfloor \frac{h}{2} \right\rfloor + 3$. Using the same pebbling as above for T_{h+1}, it is easy to see that $\mathsf{BW\text{-}Peb}^{\emptyset}(T_{h+1}) = \left\lfloor \frac{h}{2} \right\rfloor + 3$, and since the pebbling cost cannot increase by more than one when the height is increased by one we get $\mathsf{BW\text{-}Peb}^{\emptyset}(T_{h+2}) = \left\lfloor \frac{h}{2} \right\rfloor + 4 = \left\lfloor \frac{h+1}{2} \right\rfloor + 3$. In the same way we get $\mathsf{BW\text{-}Peb}(T_{h+1}) = \left\lfloor \frac{h+1}{2} \right\rfloor + 3$.

To pebble T_{h+2} in cost $\left\lfloor \frac{h+1}{2} \right\rfloor + 3$ leaving a pebble on z, first black-pebble the root z_1 of the subtree T_{h+1}^1 in cost $\left\lfloor \frac{h+1}{2} \right\rfloor + 3$. Leaving the pebble on z_1, make a pebbling visiting the root z_2 of T_{h+1}^2 in cost $\left\lfloor \frac{h}{2} \right\rfloor + 3 = \left\lfloor \frac{h+1}{2} \right\rfloor + 2$ using the pebbling for T_{h+1}^2 constructed inductively. In this pebbling there is a time t when z_2 is pebbled and T_{h+1}^2 contains at most $\left\lfloor \frac{h+1}{2} \right\rfloor + 1$ pebbles. At this time t, place a black pebble on z and remove the black pebble on z_1 without exceeding the total limit of $\left\lfloor \frac{h+1}{2} \right\rfloor + 3$ pebbles on T_{h+2}. Then finish the pebbling of T_{h+1}^2. The theorem follows. $\square$

5 Resolution and Pebbling Contradictions

A *pebbling contradiction* defined on a DAG G encodes the pebble game on G by defining the sources to be true and the targets false, and specifying that truth propagates through the graph according to the pebbling rules.

Definition 5.1 (Pebbling contradiction [12]). Let G be a DAG with sources S and a unique target z and with all vertices of G having indegree 0 or 2, and let $d \in \mathbb{N}^+$. Associate d distinct variables $x(v)_1, \ldots, x(v)_d$ with every vertex $v \in V(G)$. The dth degree *pebbling contradiction* on G, denoted Peb_G^d, is the conjunction of the following clauses:

- $\bigvee_{i=1}^{d} x(s)_i$ for all $s \in S$ (*source axioms*),
- $\overline{x(z)_i}$ for all $i \in [d]$ (*target axioms*),

$$
\begin{aligned}
&(x(r)_1 \vee x(r)_2) \\
\wedge\ &(x(s)_1 \vee x(s)_2) \\
\wedge\ &(x(t)_1 \vee x(t)_2) \\
\wedge\ &(\overline{x(r)}_1 \vee \overline{x(s)}_1 \vee x(u)_1 \vee x(u)_2) \\
\wedge\ &(\overline{x(r)}_1 \vee \overline{x(s)}_2 \vee x(u)_1 \vee x(u)_2) \\
\wedge\ &(\overline{x(r)}_2 \vee \overline{x(s)}_1 \vee x(u)_1 \vee x(u)_2) \\
\wedge\ &(\overline{x(r)}_2 \vee \overline{x(s)}_2 \vee x(u)_1 \vee x(u)_2) \\
\wedge\ &(\overline{x(s)}_1 \vee \overline{x(t)}_1 \vee x(v)_1 \vee x(v)_2) \\
\wedge\ &(\overline{x(s)}_1 \vee \overline{x(t)}_2 \vee x(v)_1 \vee x(v)_2) \\
\wedge\ &(\overline{x(s)}_2 \vee \overline{x(t)}_1 \vee x(v)_1 \vee x(v)_2) \\
\wedge\ &(\overline{x(s)}_2 \vee \overline{x(t)}_2 \vee x(v)_1 \vee x(v)_2)
\end{aligned}
$$

$$
\begin{aligned}
\wedge\ &(\overline{x(u)}_1 \vee \overline{x(v)}_1 \vee x(z)_1 \vee x(z)_2) \\
\wedge\ &(\overline{x(u)}_1 \vee \overline{x(v)}_2 \vee x(z)_1 \vee x(z)_2) \\
\wedge\ &(\overline{x(u)}_2 \vee \overline{x(v)}_1 \vee x(z)_1 \vee x(z)_2) \\
\wedge\ &(\overline{x(u)}_2 \vee \overline{x(v)}_2 \vee x(z)_1 \vee x(z)_2) \\
\wedge\ &\overline{x(z)}_1 \\
\wedge\ &\overline{x(z)}_2
\end{aligned}
$$

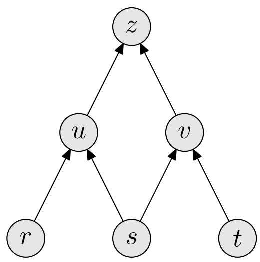


Figure 1: The pebbling contradiction $Peb^2_{\Pi_2}$ for the pyramid graph Π_2 of height 2.

- $\overline{x(u)}_i \vee \overline{x(v)}_j \vee \bigvee_{l=1}^{d} x(w)_l$ for all $i, j \in [d]$ and all $w \in V(G) \setminus S$, where u, v are the two predecessors of w (*pebbling axioms*).

The formula Peb^d_G is a $(2{+}d)$-CNF formula with $\mathrm{O}\big(d^2 \cdot \big|V(G)\big|\big)$ clauses over $d \cdot |V(G)|$ variables. See Figure 1 for an example pebbling contradiction.

It is easy to see that pebbling contradictions are unsatisfiable. Peb^d_G can be refuted in resolution by deriving $\bigvee_{i=1}^{d} x(v)_i$ for all $v \in V(G)$ inductively in topological order and then resolving with the target axioms $\overline{x(z)}_i$, $i \in [d]$. This proves the next theorem.

Theorem 5.2 ([11]). *For any DAG G with all vertices having indegree 0 or 2, there is a resolution refutation $\pi : Peb^d_G \to 0$ with $L(\pi) = \mathrm{O}\big(d^2 \cdot \big|V(G)\big|\big)$ and $W(\pi) \leq 2d$.*

Tree-like resolution is good at refuting pebbling contradictions Peb^1_G but is bad at refuting Peb^d_G for $d \geq 2$.

Theorem 5.3 ([9]). *For any DAG G with all vertices having indegree 0 or 2, there is a tree-like resolution refutation π of Peb^1_G such that $L(\pi) = \mathrm{O}\big(\big|V(G)\big|\big)$ and $Sp(\pi) = \mathrm{O}(1)$.*

Theorem 5.4 ([11]). *For any DAG G with all vertices having indegree 0 or 2, $L_{\mathfrak{T}}\big(Peb^2_G \vdash 0\big) = 2^{\Omega(\mathsf{Peb}(G))}$.*

For refutation clause space, the upper bound $Sp\big(Peb^d_G \vdash 0\big) = \mathsf{Peb}(G) + C$, where C is a constant independent of d, is fairly obvious: Just take an optimal black pebbling and derive $\bigvee_{i=1}^{d} x(v)_i$ when vertex v is pebbled. This is not quite an optimal strategy with respect to clause space, however. We can do at least a little bit better.

Theorem 5.5 ([25]). *$Sp\big(Peb^2_{T_h} \vdash 0\big) \leq \big\lceil \frac{2h+1}{3} \big\rceil + 3 = \frac{2}{3}\mathsf{Peb}(G) + \mathrm{O}(1)$.*

It is not known if this bound is tight, since no corresponding lower bound on $Sp\big(Peb^d_G \vdash 0\big)$ has been shown for pebbling degree $d \geq 2$ in unrestricted resolution (in terms of the pebbling price or otherwise). The only previously known lower bound on the refutation clause space of pebbling contradictions is a bound $Sp_{\mathfrak{T}}\big(Peb^d_{T_h} \vdash 0\big) = h + \mathrm{O}(1)$ for the special case of tree-like resolution [25]. Unfortunately, this does not tell us anything about unrestricted resolution. For tree-like resolution, if the only way of deriving D is from clauses C_1, C_2 such that $Sp_{\mathfrak{T}}\big(F \vdash C_i\big) \geq s$, then $Sp_{\mathfrak{T}}\big(F \vdash D\big) \geq s + 1$ since one of the clauses C_i must be kept in memory while deriving the other clause. This seems to be very different from how unrestricted resolution works with respect to space.

However, the resolution refutation of $Peb^2_{T_h}$ in the proof of Theorem 5.5 in [25] is structurally quite similar to the optimal black-white pebbling of T_h presented in [34], and it is hard to see how any resolution refutation could do better. This raises the suspicion that the black-white pebbling price $\mathsf{BW\text{-}Peb}(T_h) \approx h/2$ might be a lower bound for $Sp\big(Peb^d_{T_h} \vdash 0\big)$, and in general that $Sp\big(Peb^d_G \vdash 0\big) \geq \mathsf{BW\text{-}Peb}(G)$ for any DAG G and $d \geq 2$.

This suspicion is somewhat strengthened by the fact that for variable space, we do have a lower bound for unrestricted resolution.[2]

Theorem 5.6 ([9]). *For any $d \in \mathbb{N}^+$, $VarSp\big(Peb^d_G \vdash 0\big) \geq \mathsf{BW\text{-}Peb}(G)$.*

If the refutation clause space of pebbling contradictions would be constant, Theorem 5.6 would imply that as $\mathsf{BW\text{-}Peb}(G)$ grows larger, the clauses in memory get wider, and thus weaker. Still it would somehow be possible to derive a contradiction from a constant number of these clauses of unbounded width. This appears counterintuitive.

On the other hand, for $d = 1$ refutations of Peb^1_G in constant space have exactly these "counterintuitive" properties. The resolution refutation of Peb^1_G in [9] is constructed by first downloading the pebbling axiom for the target z and then propagating falsity downwards by resolving with pebbling axioms for vertices $v \in V\big(G\big) \setminus S$ in reverse topological order. This finally yields a clause $\bigvee_{v \in S} \overline{x(v)}_1 \lor x(z)_1$ of width $|S| + 1$, which can be eliminated by resolving one by one with the source axioms $x(v)_1$ for all $v \in S$ and then with the target axiom $\overline{x(z)}_1$ to yield the empty clause 0.

If we want to establish a non-constant lower bound on $Sp\big(Peb^d_G \vdash 0\big)$ for $d \geq 2$, we have to pin down why this case is different. Intuitively, the difference is that with only one variable per vertex, a single CNF clause $\overline{x(v_1)}_1 \lor \ldots \lor \overline{x(v_m)}_1$ can express the disjunction of the falsity of an arbitrary number of vertices $v_1, \ldots, v_m$, but for $d = 2$, the straightforward way of expressing that both variables $x(v_i)_1$ and $x(v_i)_2$ are false for at least one out of m vertices requires 2^m CNF clauses.

A resolution proof refutes a pebbling contradiction by deriving $x(v)_i$ and $\overline{x(v)}_i$ for some variable $x(v)_i$ and then resolving to get 0, or, in other words, by proving that some vertex v is both true and false. Arguing very informally, if we let black pebbles in a DAG G correspond to the conjunction of truth $\bigvee_{i=1}^{d} x(v)_i$ for all black-pebbled vertices v, and white pebbles in G correspond to the disjunction of falsity $\bigwedge_{i=1}^{d} \overline{x(w)}_i$ for all white-pebbled vertices w, the clauses in a pebbling contradiction encode that truth propagates "upwards" and falsity "downwards" in Peb^d_G exactly in accordance with the rules of the

[2] To be precise, the result in [9] is for $d = 1$, but the proof generalizes easily to any $d \in \mathbb{N}^+$.

black-white pebble game on G. In view of this, is does not seem too far-fetched that a resolution refutation should somehow have to mimic a pebbling of the DAG on which the formula is based.

If we could make the connection between resolution and pebbling by associating truth with black pebbles and falsity with white pebbles, for $d \geq 2$ we would expect that such a connection should yield a lower bound on the refutation space of a pebbling contradiction in terms of the pebbling price of the underlying graph. This is the guiding intuition behind the result in this paper.

6 Modifying the Black-White Pebble Game

To prove a lower bound on the refutation clause space of pebbling contradictions, we want to interpret resolution derivations in terms of pebble placements on the corresponding graph. The translation from sets of clauses to sets of pebbles, which is presented in Section 8, follows the ideas sketched at the end of the previous section, but the problem is that the pebble configurations induced by a resolution derivation using this translation do not obey the rules of the black-white pebble game. Therefore, we need to alter the pebbling rules.

In this section, we present the modified pebble game used for analyzing resolution derivations. Assuming a technical lemma, we then show that for binary trees we get essentially the same bound on pebbling price in this new pebble game as in the black-white pebble game of Definition 4.1. The rather lengthy proof of the key lemma is given in the next section.

We define our adapted pebble game in two steps. Our first modification is that in the context of resolution, it appears that a more natural rule for white pebble removal is that a white pebble can be removed from a vertex when a black pebble is placed on that same vertex. This does not really change anything.

Definition 6.1 (S-pebble game). Suppose that G is a DAG with sources S and a single target z. The *superpositioned black-white pebble game*, or *S-pebble game*, is as in Definition 4.1, except that a vertex may have both a black and a white pebble on it, and the pebbling rules are (1)–(3) in Definition 4.1 and (4') below instead of rule (4) in Definition 4.1.

4'. A white pebble on v can be removed only if there is a black pebble on v.

Lemma 6.2. *Suppose that $\mathcal{S} = \{\mathbb{S}_0, \ldots, \mathbb{S}_\tau\}$ is an S-pebbling of a DAG G. Then there is an ordinary black-white pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $W_t \subseteq W'_t$ and $B_t \subseteq B'_t$ for $\mathbb{P}_t = (B_t, W_t)$ and $\mathbb{S}_t = (B'_t, W'_t)$. In particular, $\mathsf{cost}(\mathcal{P}) \leq \mathsf{cost}(\mathcal{S})$.*

Proof sketch. The construction is by forward induction over $\mathcal{S}$. The only problem is rule (4'), but it is easy to eliminate white pebbles in $\mathcal{P}$ before they are removed by a move (4') in $\mathcal{S}$. If there was a black pebble on v in $\mathcal{S}$ when the white pebble was placed there, we ignore this white pebble placement in $\mathcal{P}$. If the white pebble was there before the black pebble, remove the white pebble in $\mathcal{P}$ when the black pebble is placed there in $\mathcal{S}$. The rest is technical details.

Note that to avoid being overly formalistic, we ignore the fact there there might be "idle moves" $\mathbb{P}_t = \mathbb{P}_{t+1}$ and moves simultaneously removing a white pebble and placing a black one in $\mathcal{P}$. It is clear that this is not a problem. $\square$

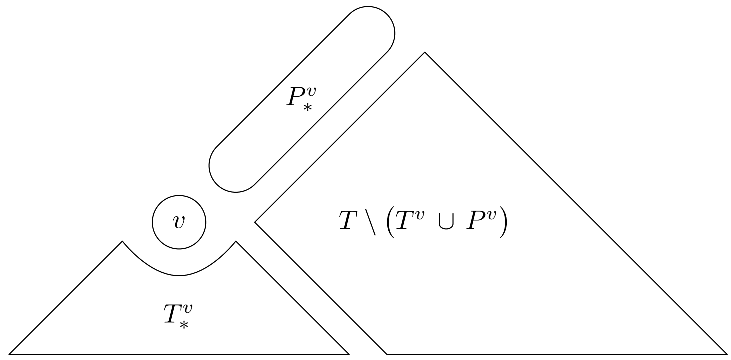


Figure 2: Referencing sets of vertices of a tree T relative to a vertex $v \in V(T)$.

Our second, and far more substantial, modification of the pebble game is motivated by the fact that when analyzing resolution derivations, we are forced to deal with "backward" pebbling moves and even "illegal erasures" of white pebbles. In order to prove lower bounds for a pebble game allowing for such moves, we have to keep track of exactly which white pebbles have been used to get a black pebble on a vertex. Loosely put, removing a white pebble from a vertex v without placing a black pebble on the same vertex should be in order, provided that all black pebbles placed on vertices above v in the DAG with the help of the white pebble on v are removed as well.

We need some notation and terminology to define and analyze our new pebble game. Recall that T denotes a complete binary tree with root z. We use p, q, r, u, v, w to denote arbitrary vertices in $V(T)$ and U, V, W to denote arbitrary subsets of vertices in $V(T)$.

For v a vertex of T, we let T^v denote the vertices in the complete binary subtree of T rooted at v, and $T_*^v = T^v \setminus \{v\}$ the vertices in T^v without its root v. We let P^v denote the vertices in the unique path from v to the root z of T and $P_*^v = P^v \setminus \{v\}$ the path without v (see Figure 2).

We say that the vertex v is *below* u if $v \in T_*^u$ and *above* u if $v \in P_*^u$. We say that u and v are *unrelated* if $v \notin T^u \cup P^u$. We let $succ(v)$ denote the immediate successor of v and $pred(v)$ the immediate predecessors. For a leaf v we have $pred(v) = \emptyset$ and for the root z we have $succ(z) = \emptyset$. If $succ(u) = succ(v)$ for $u \neq v$, u and v are *siblings*, and we write $v = sibl(u)$. We blur the distinction somewhat between a tree T and the vertices in $V(T)$ and write for instance $T \setminus (T^v \cup P^v)$ instead of $V(T) \setminus (T^v \cup P^v)$ to denote all vertices in the tree unrelated to v.

The following definition extends the concepts of *below* and *above* from vertices to sets of vertices.

Definition 6.3. For sets of vertices V, W in a binary tree, we say that W is a *roof over* V if there is no $w \in W$ such that $P_*^w \cap V \neq \emptyset$ and for each $v \in V$ there is a $w \in P^v \cap W$. The set W is *below* the vertex u if $W \subseteq T_*^u$. If $P_*^w \cap W = \emptyset$ for all $w \in W$, the vertex set W is *simple*.

Next, we present the concept used to "label" each black pebble with the set of white pebbles (if any) this black pebble is dependent on. It might be

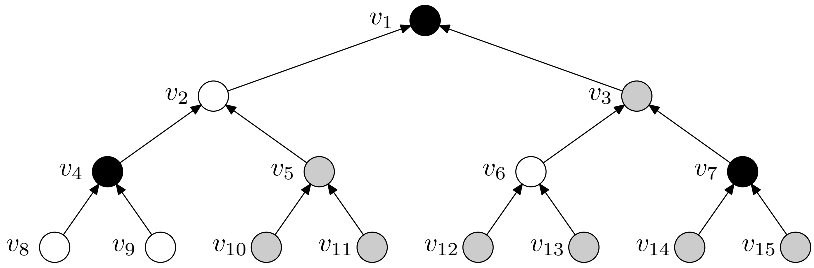


Figure 3: The pebble subconfigurations $v_1\langle v_2, v_6\rangle$, $v_4\langle v_8, v_9\rangle$ and $v_7\langle\emptyset\rangle$.

easier to parse this rather technical definition by first studying Figure 3 and the explanations in Example 6.5.

Definition 6.4 (Subconfiguration). If v is a vertex of T and $W \subseteq T^v_*$ is a simple set below v, we say that $v\langle W\rangle$ is a pebble *subconfiguration* with a black pebble on v supported by white pebbles on $w \in W$. The black pebble on v in $v\langle W\rangle$ is said to be *dependent* on the white pebbles in W. We refer to $v\langle\emptyset\rangle$ as an *independent black pebble*.

We define the *cover* of $v\langle W\rangle$ to be $\mathit{cover}(v\langle W\rangle) = T^v \setminus \bigcup_{w\in W} T^w$. The *boundary* of $v\langle W\rangle$ is $\partial v\langle W\rangle = \{v\} \cup W$. The *interior* of $v\langle W\rangle$ is $\mathit{int}(v\langle W\rangle) = \mathit{cover}(v\langle W\rangle) \setminus \partial v\langle W\rangle$ and the *closure* is $\mathit{cl}(v\langle W\rangle) = \mathit{cover}(v\langle W\rangle) \cup \partial v\langle W\rangle$.

If $\mathit{cover}(v\langle V\rangle) \subseteq \mathit{cover}(u\langle U\rangle)$, we say that $v\langle V\rangle$ is *covered by* $u\langle U\rangle$ and write $v\langle V\rangle \preceq u\langle U\rangle$. If $v\langle V\rangle \preceq u\langle U\rangle$ and $v\langle V\rangle \neq u\langle U\rangle$, we write $v\langle V\rangle \prec u\langle U\rangle$. If $\mathit{cover}(v\langle V\rangle) \cap \mathit{cover}(u\langle U\rangle) = \emptyset$, the subconfigurations $v\langle V\rangle$ and $u\langle U\rangle$ are *non-overlapping*. If $\mathit{cl}(v\langle V\rangle) \cap \mathit{cl}(u\langle U\rangle) = \emptyset$, $v\langle V\rangle$ and $u\langle U\rangle$ are *non-touching*.

When we specify the set W of white-pebbled vertices in $v\langle W\rangle$ by enumerating the members of W, we will abuse notation somewhat by omitting the curly brackets inside $\langle$ and $\rangle$ around this set.

Example 6.5. Consider the subconfigurations in Figure 3. For $v_1\langle v_2, v_6\rangle$ we have

$$\begin{aligned}
\mathit{cover}(v_1\langle v_2, v_6\rangle) &= \{v_1, v_3, v_7, v_{14}, v_{15}\},\\
\partial v_1\langle v_2, v_6\rangle &= \{v_1, v_2, v_6\},\\
\mathit{int}(v_1\langle v_2, v_6\rangle) &= \{v_3, v_7, v_{14}, v_{15}\},\\
\mathit{cl}(v_1\langle v_2, v_6\rangle) &= \{v_1, v_2, v_3, v_6, v_7, v_{14}, v_{15}\}.
\end{aligned}$$

Since $\mathit{cl}(v_4\langle v_8, v_9\rangle) = \{v_4, v_8, v_9\}$, the subconfigurations $v_1\langle v_2, v_6\rangle$ and $v_4\langle v_8, v_9\rangle$ are non-touching. For $v_7\langle\emptyset\rangle$ we have $\mathit{cover}(v_7\langle\emptyset\rangle) = \{v_7, v_{14}, v_{15}\}$, so $v_7\langle\emptyset\rangle$ and $v_1\langle v_2, v_6\rangle$ are overlapping, or more precisely it holds that $v_7\langle\emptyset\rangle \prec v_1\langle v_2, v_6\rangle$.

Note that $\preceq$ is an order relation and that the minimal elements are pebble subconfigurations $v\langle \mathit{pred}(v)\rangle$. We will use the following characterization of $\preceq$ repeatedly.

Observation 6.6. *$v\langle V\rangle \preceq u\langle U\rangle$ if and only if $v \in T^u$, $P^v \cap U = \emptyset$ and V is a simple roof below v over $U \cap T^v$.*

Proof. By Definition 6.4, U and V are simple sets below u and v, respectively, and $v\langle V\rangle \preceq u\langle U\rangle$ if and only if $\mathit{cover}(v\langle V\rangle) = T^v \setminus \bigcup_{w\in V} T^w \subseteq T^u \setminus \bigcup_{w\in U} T^u = \mathit{cover}(u\langle U\rangle)$.

($\Rightarrow$) Suppose that $\mathit{cover}(v\langle V\rangle) \subseteq \mathit{cover}(u\langle U\rangle)$. Since $v \in \mathit{cover}(v\langle V\rangle) \subseteq \mathit{cover}(u\langle U\rangle)$, we have $v \in T^u$ but $v \notin \bigcup_{w\in U} T^w$, and this second condition is equivalent to $P^v \cap U = \emptyset$. If V is not a roof over $U \cap T^v$, there is a $w \in U \cap T^v$ such that $P^w \cap V = \emptyset$. For such a w we would have $w \in \mathit{cover}(v\langle V\rangle)$ but $w \notin \mathit{cover}(u\langle U\rangle)$, which contradicts $\mathit{cover}(v\langle V\rangle) \subseteq \mathit{cover}(u\langle U\rangle)$.

($\Leftarrow$) Suppose that $v \in T^u$ and $P^v \cap U = \emptyset$ for V a simple roof below v over $U \cap T^v$, but that $\mathit{cover}(v\langle V\rangle) \not\subseteq \mathit{cover}(u\langle U\rangle)$. By assumption $T^v \subseteq T^u$ and $v \notin \bigcup_{w\in U} T^w$, so $v \in \mathit{cover}(u\langle U\rangle)$. Hence, there must exist a $v' \in T^v_*$ such that $v' \in \mathit{cover}(v\langle V\rangle) \setminus \mathit{cover}(u\langle U\rangle)$ and $\mathit{succ}(v') \in \mathit{cover}(u\langle U\rangle)$. This implies that $v' \in U \cap T^v$, but the fact that $v' \in \mathit{cover}(v\langle V\rangle)$ shows that $P^{v'} \cap V = \emptyset$. That is, V is not a roof over $U \cap T^v$. Contradiction. $\square$

Our modified black-white pebble game is defined in terms of subconfigurations of black- and white-pebbled vertices.

Definition 6.7 (Labelled black-white pebble game). For T a binary tree with root z, a *labelled black-white pebbling*, or *L-pebbling*, on T is a sequence $\mathcal{L} = \{\mathbb{L}_0 = \{\emptyset\}, \mathbb{L}_1, \ldots, \mathbb{L}_\tau\}$ of sets of subconfigurations $\mathbb{L}_t$ such that $\mathbb{L}_{t+1}$ is obtained from $\mathbb{L}_t$ by one of the following rules:

Introduction $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \{v\langle \mathit{pred}(v)\rangle\}$ for $v\langle \mathit{pred}(v)\rangle \notin \mathbb{L}_t$.

Merger $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \{u\langle (U \cup V) \setminus \{v\}\rangle\}$ for $u\langle U\rangle$, $v\langle V\rangle \in \mathbb{L}_t$ such that $v \in U$.

Reversal $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \{v\langle V\rangle\}$ if $v\langle V\rangle \prec u\langle U\rangle$ for some $u\langle U\rangle \in \mathbb{L}_t$.

Erasure $\mathbb{L}_{t+1} = \mathbb{L}_t \setminus \{v\langle V\rangle\}$ for $v\langle V\rangle \in \mathbb{L}_t$.

A *legal L-pebbling of T* is an L-pebbling $\mathcal{L}$ such that $\mathbb{L}_\tau = \{z\langle\emptyset\rangle\}$. We write $u\langle U\rangle = \mathsf{merge}(v\langle V\rangle, w\langle W\rangle)$ if $u\langle U\rangle = v\langle (V \cup W) \setminus \{w\}\rangle$ for $w \in V$, and refer to this as a *merger on w*.

Let $\mathit{Bl}(\mathbb{L}_t) = \{v \in V(T) \mid \exists\, v\langle W\rangle \in \mathbb{L}_t\}$ denote the black pebbles in $\mathbb{L}_t$ and $\mathit{Wh}(\mathbb{L}_t) = \{w \in V(T) \mid \exists v\langle W\rangle \in \mathbb{L}_t \text{ s.t. } w \in W\}$ the white pebbles. The *cost* of a set of subconfigurations $\mathbb{L}$ is $\mathsf{cost}(\mathbb{L}) = |\mathit{Bl}(\mathbb{L}) \cup \mathit{Wh}(\mathbb{L})|$. The cost of an L-pebbling $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ is $\max_{t\in[\tau]}\{\mathsf{cost}(\mathbb{L}_t)\}$. The *L-pebbling price* $\mathsf{L\text{-}Peb}(v\langle W\rangle)$ of a subconfiguration $v\langle W\rangle$ is the minimum cost of any L-pebbling such that $\mathbb{L}_\tau = \{v\langle W\rangle\}$, and the L-pebbling price of the binary tree T is $\mathsf{L\text{-}Peb}(T) = \mathsf{L\text{-}Peb}(z\langle\emptyset\rangle)$.

In the L-pebble game, one can remove a white pebble without placing a black pebble on the same vertex, but if so the rule for erasure makes sure that any black pebble dependent on the removed white pebble is removed as well. A normal removal of a white pebble from w according to rule (4’) of the S-pebble game corresponds to merging $v\langle V\rangle$ and $w\langle W\rangle$ into $v\langle (V \cup W) \setminus \{w\}\rangle$ and then erasing $v\langle V\rangle$ and $w\langle W\rangle$. Note that if $u\langle U\rangle = \mathsf{merge}(v\langle V\rangle, w\langle W\rangle)$, then $\mathit{cover}(u\langle U\rangle) = \mathit{cover}(v\langle V\rangle) \mathbin{\dot\cup} \mathit{cover}(w\langle W\rangle)$, where $\dot\cup$ denotes disjoint union.

The “backward” pebbling moves mentioned in the beginning of this section are moves according to the reversal rule. For an ordinary black-white pebbling of a binary tree, $\mathit{cover}(v\langle V\rangle)$ is the set of vertices already taken care of by $v\langle V\rangle$ in the sense that if the rest of the pebbling is performed optimally no black pebbles will be placed on $\mathit{cover}(v\langle V\rangle)$. Consequently, in an optimal black-white pebbling we will never move from a subconfiguration $u\langle U\rangle$ to some $v\langle V\rangle \prec u\langle U\rangle$.

Arguing very informally, it seems plausible that making reversals in an L-pebbling should only "weaken" the pebble configurations (for example, reversing from $v_1\langle v_2, v_6\rangle$ to $v_7\langle\emptyset\rangle$ in Figure 3), and so adding the rule for reversal should not decrease the pebbling cost.

For our purposes, it is sufficient to prove that the L-pebbling price of a binary tree T is *asymptotically* at least as large as the ordinary black-white pebbling price $\mathsf{BW\text{-}Peb}(T)$. The main obstacle in the proof is how to handle the reversal moves. In view of the informal argument above, it might seem intuitively clear that an optimal L-pebbling strategy does not need any reversal moves. Unfortunately, proving that reversal moves do not affect the asymptotical pebbling price turns out to be rather involved. And in fact, if we generalize the L-pebble game from trees to arbitrary DAGs in the natural way, the statement is even false. For instance, it fails for pyramid graphs Π_h (Figure 1 on page 13). Klawe [29] showed that $\mathsf{BW\text{-}Peb}(\Pi_h) = h/2 + \mathrm{O}(1)$, but it is not too hard to see that Π_h can be L-pebbled with $\mathrm{O}(1)$ pebbles if we allow moving black pebbles downwards.

What we need to get rid of reversal moves is Lemma 6.8 stated below. We spend the rest of this section demonstrating how the desired lower bound $\mathsf{L\text{-}Peb}(T) = \Omega\bigl(\mathsf{BW\text{-}Peb}(T)\bigr)$ follows from this assumption, postponing a proof of Lemma 6.8 until the next section.

Lemma 6.8. *Suppose that $\mathcal{L}$ is an L-pebbling of a complete binary tree T. Then there is an L-pebbling $\mathcal{L}'$ of T without reversals such that $\mathsf{cost}(\mathcal{L}') = \mathrm{O}\bigl(\mathsf{cost}(\mathcal{L})\bigr)$.*

It is not too hard to see that taking a legal reversal-free L-pebbling $\mathcal{L} = \bigl\{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\bigr\}$ of T and looking at $\bigl\{Bl\bigl(\mathbb{L}_t\bigr), Wh\bigl(\mathbb{L}_t\bigr)\bigr\}$ for $1 \le t \le \tau$, we get a legal S-pebbling of T in at most the same cost. We prove this formally in the next two lemmas.

Lemma 6.9. *Suppose that $\mathcal{L}$ is a reversal-free L-pebbling of T. Then there is a reversal-free L-pebbling $\mathcal{L}'$ of T with $\mathsf{cost}(\mathcal{L}') \le \mathsf{cost}(\mathcal{L})$ such that every $v\langle V\rangle$ in $\mathcal{L}'$ occurs during one contiguous time interval, and every $v\langle V\rangle$ in $\mathcal{L}'$ except $z\langle\emptyset\rangle$ is used in exactly one merger, after which it is erased.*

Proof. We construct $\mathcal{L}'$ by backward induction over $\mathcal{L} = \bigl\{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\bigr\}$. Let $\mathbb{L}'_\tau = \mathbb{L}_\tau = \bigl\{z\langle\emptyset\rangle\bigr\}$. Our induction hypothesis is that $\mathbb{L}'_t \subseteq \mathbb{L}_t$ for $\mathbb{L}'_t$ consisting of non-overlapping subconfigurations. The backward induction step from $t+1$ to t is a case analysis over the moves $\mathbb{L}_t \rightsquigarrow \mathbb{L}_{t+1}$ in $\mathcal{L}$.

Introduction $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \bigl\{v\langle pred(v)\rangle\bigr\}$: Set $\mathbb{L}'_t = \mathbb{L}'_{t+1} \setminus \bigl\{v\langle pred(v)\rangle\bigr\}$. Note that we might have $\mathbb{L}'_t = \mathbb{L}'_{t+1}$ if $v\langle pred(v)\rangle \notin \mathbb{L}'_{t+1}$. In any case, the induction hypothesis holds for $\mathbb{L}'_t$.

Merger $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \bigl\{v\langle(V \cup W) \setminus \{w\}\rangle\bigr\}$: If $v\langle(V \cup W) \setminus \{w\}\rangle \notin \mathbb{L}'_{t+1}$, set $\mathbb{L}'_t = \mathbb{L}'_{t+1}$. The induction hypothesis trivially remains true. Otherwise, set $\mathbb{L}'_t = \bigl(\mathbb{L}'_{t+1} \cup \bigl\{v\langle V\rangle, w\langle W\rangle\bigr\}\bigr) \setminus \bigl\{v\langle(V \cup W) \setminus \{w\}\rangle\bigr\}$. By the induction hypothesis we have $v\langle V\rangle, w\langle W\rangle \notin \mathbb{L}'_{t+1}$, since $\mathbb{L}'_{t+1}$ is non-overlapping and $v\langle V\rangle$ and $w\langle W\rangle$ are covered by $\mathsf{merge}(v\langle V\rangle, w\langle W\rangle)$ by Definitions 6.4 and 6.7. For the same reason $\mathbb{L}'_t$ must be non-overlapping. We can get from $\mathbb{L}'_t$ to $\mathbb{L}'_{t+1}$ in three steps $\mathbb{L}'_{t+1/3} = \mathbb{L}'_t \cup \bigl\{v\langle(V \cup W) \setminus \{w\}\rangle\bigr\}$, $\mathbb{L}'_{t+2/3} = \mathbb{L}'_{t+1/3} \setminus \bigl\{v\langle V\rangle\bigr\}$, $\mathbb{L}'_{t+1} = \mathbb{L}'_{t+2/3} \setminus \bigl\{w\langle W\rangle\bigr\}$ by first merging $v\langle V\rangle$ and $w\langle W\rangle$, then erasing $v\langle V\rangle$ and finally erasing $w\langle W\rangle$.

Erasure $\mathbb{L}_{t+1} = \mathbb{L}_t \setminus \{v\langle V\rangle\}$: All erasure moves in $\mathcal{L}'$ are taken care of in connection with mergers, so set $\mathbb{L}'_t = \mathbb{L}'_{t+1}$.

We claim that all moves in $\mathcal{L}'$ constructed in this way are legal. If $u\langle U\rangle \in \mathbb{L}'_t$, then $u\langle U\rangle \in \mathbb{L}_t$ and for $u\langle U\rangle \neq u\langle pred(u)\rangle$ we know that this subconfiguration must have been derived at a time $t' \leq t$ in $\mathcal{L}$ by a merger of $v\langle V\rangle, w\langle W\rangle \prec u\langle U\rangle$. Thus the backward construction of $\mathcal{L}'$ will yield a correct derivation of $u\langle U\rangle$.

Also, any subconfiguration $v\langle V\rangle$ occurs only in one merger, after which it is immediately erased. At all times $t' > t$ after which $v\langle V\rangle$ was erased from $\mathcal{L}'$ directly after the first merger move, there is a $u\langle U\rangle \succ v\langle V\rangle$ in $\mathbb{L}'_{t'}$. Since all $\mathbb{L}'_{t'}$ are non-overlapping, the subconfiguration $v\langle V\rangle$ never appears again (this can easily be formalized by a forward induction argument).

Finally, by construction $\mathbb{L}'_t \subseteq \mathbb{L}_t$, and for the merger moves in $\mathcal{L}'$ we have $\mathbb{L}'_{t+1/3}, \mathbb{L}'_{t+2/3} \subseteq \mathbb{L}_{t+1}$. This shows that for all $\mathbb{L}' \in \mathcal{L}'$, there is a corresponding $\mathbb{L} \in \mathcal{L}$ such that $\mathsf{cost}(\mathbb{L}') \leq \mathsf{cost}(\mathbb{L})$, and it follows that $\mathsf{cost}(\mathcal{L}') \leq \mathsf{cost}(\mathcal{L})$. ☐

Lemma 6.10. *Suppose that $\mathcal{L}$ is a reversal-free L-pebbling of a complete binary tree T. Then there is an S-pebbling $\mathcal{S}$ of T such that $\mathsf{cost}(\mathcal{S}) \leq \mathsf{cost}(\mathcal{L})$.*

Proof. By Lemma 6.9, without loss of generality we can assume that each $v\langle V\rangle$ is erased from $\mathcal{L}$ precisely after it has been used in a merger, and that $v\langle V\rangle$ is erased before $w\langle W\rangle$ when both subconfigurations are eliminated after a move $v\langle (V \cup W) \setminus \{w\}\rangle = \mathsf{merge}(v\langle V\rangle, w\langle W\rangle)$, so that the white pebble on w is removed before the black pebble on w.

It is clear that we are done if we can construct an S-pebbling $\mathcal{S}$ with moves matching the moves in $\mathcal{L}$ exactly. Let $\mathbb{S}_0 = (\emptyset, \emptyset)$ and construct $\mathbb{S}_{t+1}$ inductively by looking at the moves in $\mathbb{L}_t \rightsquigarrow \mathbb{L}_{t+1}$.

Introduction $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \{v\langle pred(v)\rangle\}$: Place white pebbles on $pred(v)$ and then a black pebble on v in $\mathcal{S}$.

Merger $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \{v\langle (V \cup W) \setminus \{w\}\rangle\}$ for $v\langle V\rangle$, $w\langle W\rangle \in \mathbb{L}_t$: No pebbling moves in $\mathcal{S}$, but note that if $v\langle V\rangle$ is now removed, the change in pebbles on T in $\mathcal{L}$ is exactly the same as after an application of rule (4') on w.

Erasure $\mathbb{L}_{t+1} = \mathbb{L}_t \setminus \{v\langle V\rangle\}$: This is the only nontrivial case. In general, an erasure move in an L-pebbling can remove an arbitrary number of white pebbles without any black pebbles being even close to these white pebbles, and there is no way we can match such a move in an S-pebbling. But since we can assume that $\mathcal{L}$ is an L-pebbling as described in Lemma 6.9, we know that $v\langle V\rangle$ has just been used in a merger. It follows that the only pebble that disappears when going from $\{Bl(\mathbb{L}_t), Wh(\mathbb{L}_t)\}$ to $\{Bl(\mathbb{L}_{t+1}), Wh(\mathbb{L}_{t+1})\}$ is either the black pebble on v, which is always a legal removal, or some white pebble on $w \in V$ which has just been eliminated in the merger move by a black pebble, and this is a legal removal according to rule (4').

We see that $\mathcal{S}$ generated in this way is a legal S-pebbling, if we modify each introduction step into three pebble placement moves. ☐

Putting it all together, we get that the L-pebbling price and the black-white pebbling price of a binary tree coincide asymptotically.

Theorem 6.11. *Let T_h denote a complete binary tree of height h. Then it holds that $\textsf{L-Peb}(T_h) = \Theta\bigl(\textsf{BW-Peb}(T_h)\bigr) = \Theta(h)$.*

Proof. Clearly $\textsf{L-Peb}(T) = \mathrm{O}\bigl(\textsf{BW-Peb}(T)\bigr)$, since an L-pebbling can imitate an optimal black pebbling in cost $\textsf{Peb}(T) = \mathrm{O}\bigl(\textsf{BW-Peb}(T)\bigr)$.

In the other direction, let $\mathcal{L}$ be an arbitrary L-pebbling of T. Assuming Lemma 6.8, there exists an L-pebbling $\mathcal{L}'$ of T without reversal moves such that $\textsf{cost}(\mathcal{L}') = \mathrm{O}\bigl(\textsf{cost}(\mathcal{L})\bigr)$. By Lemma 6.10, we can construct an S-pebbling $\mathcal{S}$ of T for which $\textsf{cost}(\mathcal{S}) \leq \textsf{cost}(\mathcal{L}')$. Finally, using Lemma 6.2 we get a plain old black-white pebbling $\mathcal{P}$ of T such that $\textsf{cost}(\mathcal{P}) \leq \textsf{cost}(\mathcal{S})$. Hence $\textsf{BW-Peb}(T) = \mathrm{O}\bigl(\textsf{L-Peb}(T)\bigr)$, and the theorem follows. ☐

7 Getting Rid of Reversal Pebbling Moves

We now prove Lemma 6.8, i.e., that the reversal rule can be eliminated from the L-pebble game without increasing the pebbling price by more than a constant factor. This provides the missing link in the proof of Theorem 6.11.

Although this section is *very* technical, the structure of the argument is quite straightforward. Before plunging into the proof, we try to give an informal overview of where we are going.

1. First we show that without loss of generality we can assume that an optimal L-pebbling $\mathcal{L}$ is *non-overlapping*, by which we mean that all subconfigurations in $\mathbb{L}_t \in \mathcal{L}$ are non-overlapping with exception for those involved in the current merger or reversal move (Definition 7.6 and Lemma 7.9).

2. Then we observe that if we restrict an L-pebbling to a subset of the vertices in T in the natural way, we get a valid L-pebbling on this subset. We refer to this restriction operation as *projection* (Definition 7.7 and Proposition 7.10).

3. This leads to the idea of trying to get rid of reversals in the following way: When the cover of a set of subconfigurations $\mathbb{L}$ shrinks as the result of a reversal move, we can eliminate this reversal by projecting the L-pebbling moves made so far on what remains *after* the reversal move. If we do this by forward induction for all reversal moves in $\mathcal{L}$, we get a reversal-free L-pebbling $\mathcal{L}'$.

4. The problem is that these projection operations do not preserve pebbling cost—the pebbling $\mathcal{L}$ may contain reversal moves such that the projected pebbling $\mathcal{L}'$ becomes more expensive than $\mathcal{L}$. We identify which kind of reversals in $\mathcal{L}$ spoil our construction of a reversal-free and cheap pebbling $\mathcal{L}'$ by projection. Allowing some temporary wishful thinking, we then establish that if such *wasteful* reversals could somehow be avoided, the construction sketched above would work (Definition 7.12, Lemma 7.13 and Corollary 7.14).

5. Finally, we prove that wasteful reversals can be eliminated. If a pebbling $\mathcal{L}$ makes a wasteful reversal, we can replace such a move by a stronger, non-wasteful reversal without increasing the total pebbling cost by more than a constant factor (Lemma 7.19).

Summing this up, Lemma 6.8 follows.

The rest of this section contains the formal proof of the lemma. Although the technical machinery might appear cumbersome, we believe that the argument is followed more easily if the reader tries to digest what the definitions mean and what is proven about them simply by drawing a binary tree of suitable height and working out small examples in this binary tree while reading.

Below,we assume without loss of generality that no obviously redundant pebbling moves are performed, in the sense that if a subconfiguration $v\langle V\rangle$ is derived at time t, then this subconfiguration is not just thrown away but is used at some time $t' > t$ further on in the pebbling before being erased. We state this formally.

Observation 7.1. *Let $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ be an arbitrary L-pebbling. Then there is a pebbling $\mathcal{L}' = \{\mathbb{L}'_0, \ldots, \mathbb{L}'_{\tau'}\}$ such that $\mathsf{cost}(\mathcal{L}') \leq \mathsf{cost}(\mathcal{L})$, $\mathbb{L}'_{\tau'} = \mathbb{L}_\tau$ and if $v\langle V\rangle \in \mathbb{L}'_t \setminus \mathbb{L}'_{t-1}$, then $v\langle V\rangle$ is used in a merger or reversal move before being erased from $\mathcal{L}'$ at some time $t' > t$.*

Proof sketch. $\mathcal{L}'$ can be constructed by backward induction over $\mathcal{L}$ in the same manner as in the proof of Lemma 6.9 on page 19. □

We start by extending Definition 6.4 to sets of pebble subconfigurations, or *L-configurations*.

Definition 7.2 (L-configuration). An *L-configuration* is a set of pebble subconfigurations $\mathbb{L} = \{v_i\langle V_i\rangle \mid i \in [m]\}$.

We define $\mathit{cover}(\mathbb{L}) = \bigcup_{v_i\langle V_i\rangle \in \mathbb{L}} \mathit{cover}(v_i\langle V_i\rangle)$. We say that $\mathbb{L}_1$ is *covered by* $\mathbb{L}_2$ and write $\mathbb{L}_1 \preceq \mathbb{L}_2$ if $\mathit{cover}(\mathbb{L}_1) \subseteq \mathit{cover}(\mathbb{L}_2)$. If $\mathit{cover}(\mathbb{L}_1) = \mathit{cover}(\mathbb{L}_2)$, we say that $\mathbb{L}_1$ and $\mathbb{L}_2$ *coincide* and write $\mathbb{L}_1 \sim \mathbb{L}_2$. $\mathbb{L}$ is *non-overlapping* if all distinct $v\langle V\rangle, u\langle U\rangle \in \mathbb{L}$ are pairwise non-overlapping and *non-touching* if all distinct $v\langle V\rangle, u\langle U\rangle \in \mathbb{L}$ are pairwise non-touching. Two L-configurations $\mathbb{L}_1, \mathbb{L}_2$ are *mutually non-overlapping* or *mutually non-touching* if all $v\langle V\rangle \in \mathbb{L}_1$ and $u\langle U\rangle \in \mathbb{L}_2$ are pairwise non-overlapping or non-touching, respectively.

For an arbitrary set of vertices $V \subseteq V(T)$, we define the *canonical representation* $\mathsf{canon}(V)$ of V to be the unique non-touching $\mathbb{L}'$ such that $\mathit{cover}(\mathbb{L}') = V$. For $\mathbb{L}$ an arbitrary L-configuration, we define $\mathsf{canon}(\mathbb{L})$ to be the canonical representation of $\mathit{cover}(\mathbb{L})$. For $\mathbb{L}$ with $\mathsf{canon}(\mathbb{L}) = \mathbb{L}'$, the *boundary* of $\mathbb{L}$ is defined as $\partial\mathbb{L} = \bigcup_{v\langle V\rangle \in \mathbb{L}'} \partial v\langle V\rangle$, the *interior* is $\mathit{int}(\mathbb{L}) = \bigcup_{v\langle V\rangle \in \mathbb{L}'} \mathit{int}(v\langle V\rangle)$ and the *closure* is $\mathit{cl}(\mathbb{L}) = \bigcup_{v\langle V\rangle \in \mathbb{L}'} \mathit{cl}(v\langle V\rangle)$.

Example 7.3. Returning to Figure 3 on page 17, if we look at the L-configuration $\mathbb{L} = \{v_1\langle v_2, v_6\rangle, v_4\langle v_8, v_9\rangle, v_7\langle\emptyset\rangle\}$ we have $\mathit{cover}(\mathbb{L}) = \{v_1, v_3, v_4, v_7, v_{14}, v_{15}\}$. Since $v_7\langle\emptyset\rangle$ is covered by $v_1\langle v_2, v_6\rangle$ and the subconfigurations $v_1\langle v_2, v_6\rangle$ and $v_4\langle v_8, v_9\rangle$ are non-touching, we get the canonical representation simply by leaving out $v_7\langle\emptyset\rangle$, i.e., $\mathsf{canon}(\mathbb{L}) = \{v_1\langle v_2, v_6\rangle, v_4\langle v_8, v_9\rangle\}$. Using this canonical representation of $\mathbb{L}$, we see that

$$\begin{aligned}
\partial\mathbb{L} &= \{v_1, v_2, v_4, v_6, v_8, v_9\}, \\
\mathit{int}(\mathbb{L}) &= \{v_3, v_7, v_{14}, v_{15}\}, \\
\mathit{cl}(\mathbb{L}) &= \{v_1, v_2, v_3, v_4, v_6, v_7, v_8, v_9, v_{14}, v_{15}\}.
\end{aligned}$$

The L-configuration $\mathbb{L}$ is overlapping because of $v_7\langle\emptyset\rangle$ and $v_1\langle v_2, v_6\rangle$, but for instance $\mathbb{L}_1 = \{v_1\langle v_2, v_6\rangle, v_7\langle\emptyset\rangle\}$ and $\mathbb{L}_2 = \{v_4\langle v_8, v_9\rangle\}$ are mutually non-touching.

We allow a mild abuse of notation by omitting curly brackets around singleton L-configurations, writing for instance $v\langle V\rangle \preceq \mathbb{L}$, $u\langle U\rangle = \mathbb{L}$ and $w\langle W\rangle \cup \mathbb{L}$ instead of $\{v\langle V\rangle\} \preceq \mathbb{L}$, $\{u\langle U\rangle\} = \mathbb{L}$ and $\{w\langle W\rangle\} \cup \mathbb{L}$.

An alternative constructive definition of canonical representation is given in the following observation. We leave it to the reader to verify that the two descriptions of canonical representation are indeed equivalent.

Observation 7.4. *The canonical representation of V can be constructed as follows: for each $v \in V$ such that $succ(v) \notin V$ or $v = z$, add the subconfiguration $v\langle W\rangle$, where $W \subseteq T_*^v$ is the maximal set such that for all $w \in W$ it holds that $P_*^w \setminus P_*^v \subseteq V$ but $w \notin V$.*

As a final preliminary before moving on to part 1 in the proof outline above, we collect some properties of the L-pebbling cost function of Definition 6.7.

Proposition 7.5. *Suppose that $\mathbb{L}, \mathbb{L}_1, \ldots, \mathbb{L}_m$ are arbitrary L-configurations.*

1. *If $\mathbb{L}_1 \subseteq \mathbb{L}_2$ then $\mathsf{cost}(\mathbb{L}_1) \leq \mathsf{cost}(\mathbb{L}_2)$.*
2. *$\mathsf{cost}(\mathbb{L}_1 \cup \mathbb{L}_2) \leq \mathsf{cost}(\mathbb{L}_1) + \mathsf{cost}(\mathbb{L}_2)$.*
3. *If $\mathbb{L}$ is non-touching, $\mathsf{cost}(\mathbb{L}) = |Bl(\mathbb{L})| + |Wh(\mathbb{L})| = |\partial\mathbb{L}|$.*
4. *If $\mathbb{L}_i$ and $\mathbb{L}_j$ are mutually non-touching for $1 \leq i < j \leq m$, it holds that $\mathsf{cost}\big(\bigcup_{i=1}^m \mathbb{L}_i\big) = \sum_{i=1}^m \mathsf{cost}(\mathbb{L}_i)$.*
5. *If $\mathbb{L}'_i = \mathsf{canon}(\mathbb{L}_i)$ for $i = 1, \ldots, m$, then $\mathsf{cost}\big(\bigcup_{i=1}^m \mathbb{L}'_i\big) \leq \mathsf{cost}\big(\bigcup_{i=1}^m \mathbb{L}_i\big)$.*
6. *If $\mathbb{L}' = \mathsf{canon}(\mathbb{L})$, then $\mathsf{cost}(\mathbb{L} \cup \mathbb{L}') = \mathsf{cost}(\mathbb{L})$, and there is an L-pebbling from $\mathbb{L}$ to $\mathbb{L}'$ which does not cost more than $\mathbb{L}$.*

Proof. According to Definition 6.7, if $Bl(\mathbb{L}_1) \cup Wh(\mathbb{L}_1) \subseteq Bl(\mathbb{L}_2) \cup Wh(\mathbb{L}_2)$ then $\mathsf{cost}(\mathbb{L}_1) = |Bl(\mathbb{L}_1) \cup Wh(\mathbb{L}_1)| \leq |Bl(\mathbb{L}_2) \cup Wh(\mathbb{L}_2)| = \mathsf{cost}(\mathbb{L}_2)$.

Part 1 follows immediately from this observation. Part 2 also follows easily, since each pebble on the left-hand side is counted at least once on the right-hand side.

For part 3, using Definition 7.2 we see that if $\mathbb{L}$ is non-touching it holds that $Bl(\mathbb{L}) \cap Wh(\mathbb{L}) = \emptyset$. And if $\mathbb{L}_i$ and $\mathbb{L}_j$ are mutually non-touching we have $\big(Bl(\mathbb{L}_i) \cup Wh(\mathbb{L}_i)\big) \cap \big(Bl(\mathbb{L}_j) \cup Wh(\mathbb{L}_j)\big) = \emptyset$, which shows that each pebble on the left-hand side of part 4 is counted exactly once on the right-hand side.

Part 5 is again immediate since it is easy to show that $Bl(\mathbb{L}'_i) \subseteq Bl(\mathbb{L}_i)$ and $Wh(\mathbb{L}'_i) \subseteq Wh(\mathbb{L}_i)$ for $\mathbb{L}'_i = \mathsf{canon}(\mathbb{L}_i)$.

For part 6, $Bl(\mathbb{L} \cup \mathbb{L}') = Bl(\mathbb{L})$ and $Wh(\mathbb{L} \cup \mathbb{L}') = Wh(\mathbb{L})$, so the cost is the same. To get the claim about pebbling, note that if $v\langle V\rangle$ and $u\langle U\rangle$ are touching but non-overlapping, we can derive $w\langle W\rangle$ such that $cover(w\langle W\rangle) = cover(v\langle V\rangle) \cup cover(u\langle U\rangle)$ simply by merging $v\langle V\rangle$ and $u\langle U\rangle$. Suppose that $v\langle V\rangle$ and $u\langle U\rangle$ are overlapping for $v \in T^u$ but $v\langle V\rangle \not\preceq u\langle U\rangle$. Then we can derive $w\langle W\rangle$ with $cover(w\langle W\rangle) = cover(v\langle V\rangle) \cup cover(u\langle U\rangle)$ and substitute it for $v\langle V\rangle$ and $u\langle U\rangle$ at no extra cost by first deriving $u_i\langle V \cap T_*^{u_i}\rangle$ for all $u_i \in U \cap int(v\langle V\rangle)$ from $v\langle V\rangle$ by reversals, and then merging all $u_i\langle V \cap T_*^{u_i}\rangle$ in turn with $u\langle U\rangle$. The resulting L-configuration $\mathbb{L} \cup \{u_i\langle V \cap T_*^{u_i}\rangle\} \cup w\langle W\rangle$ costs no more than $\mathbb{L}$, since the only change is that already white-pebbled vertices are also black-pebbled. Finally, erase $u\langle U\rangle$, $v\langle V\rangle$ and all $u_i\langle V \cap T_*^{u_i}\rangle$. Repeating this for all mutually touching subconfigurations, the claim follows. ∎

Parts 5 and 6 of Proposition 7.5 tell us that for any given set of vertices, the cheapest way of covering these vertices is to use canonical L-configurations, and if $\mathbb{L}$ is not canonical, it does not cost anything extra to make $\mathbb{L}$ canonical by applying reversals and mergers followed by erasures. We define *non-overlapping pebblings* as L-pebblings which always keep the L-configurations canonical in this way. In a non-overlapping pebbling, each introduction is immediately followed by a merger when possible, each merger is immediately followed by erasures of the merged subconfigurations, and all reversals from a subconfiguration $u\langle U\rangle$ are performed in sequence after which $u\langle U\rangle$ is erased. We refer to these merger-and-erasures and reversals-and-erasure moves as *expansions* and *implosions*, respectively.

Definition 7.6 (Non-overlapping pebbling). A *non-overlapping L-pebbling* $\mathcal{L}$ is a sequence of the following types of moves.

Introduction $\mathbb{L}_{t+1} = \mathbb{L}_t \cup v\langle pred(v)\rangle$, for $v\langle pred(v)\rangle \not\preceq \mathbb{L}_t$ and $\mathbb{L}_t$ non-touching.

Expansion $\mathbb{L}_{t+3} = \big(\mathbb{L}_t \cup \mathsf{merge}(u\langle U\rangle, v\langle V\rangle)\big) \setminus \big\{u\langle U\rangle, v\langle V\rangle\big\}$ for $u\langle U\rangle$, $v\langle V\rangle \in \mathbb{L}_t$ and $\mathbb{L}_t$ non-overlapping.

Implosion $\mathbb{L}_{t+m+1} = \big(\mathbb{L}_t \setminus u\langle U\rangle\big) \cup \mathbb{M}$ for $\mathbb{L}_t$ and $\mathbb{M} = \big\{v_i\langle V_i\rangle \mid i \in [m]\big\}$ non-touching, and $\mathbb{M} \preceq u\langle U\rangle \in \mathbb{L}_t$.

We say that $u\langle U\rangle \rightsquigarrow \mathbb{M}$ is a *nontrivial implosion* if $\mathbb{M} \prec u\langle U\rangle$.

Note that after introduction and expansion the resulting L-configuration is non-overlapping, and after implosion $\mathbb{L}_{t+m+1}$ is non-touching.

We want to prove that without loss of generality we can assume L-pebblings to be non-overlapping. The notation in the proof of this fact is simplified by introducing *projections*.

Definition 7.7 (Projection). Let $u\langle U\rangle, v\langle V\rangle$ be arbitrary subconfigurations, $\mathbb{L}$ an arbitrary L-configuration, and $\mathbb{M}$ an arbitrary non-touching L-configuration.

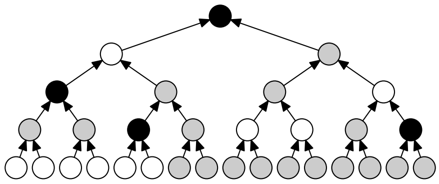
If $u\langle U\rangle$ and $v\langle V\rangle$ are overlapping, the *projection* of $u\langle U\rangle$ on $v\langle V\rangle$ is defined as $\mathsf{proj}_{v\langle V\rangle}(u\langle U\rangle) = \mathsf{canon}(\mathit{cover}(u\langle U\rangle) \cap \mathit{cover}(v\langle V\rangle))$, i.e., the unique subconfiguration $w\langle W\rangle$ such that $\mathit{cover}(w\langle W\rangle) = \mathit{cover}(u\langle U\rangle) \cap \mathit{cover}(v\langle V\rangle)$. If $u\langle U\rangle$ and $v\langle V\rangle$ are non-overlapping, we define $\mathsf{proj}_{v\langle V\rangle}(u\langle U\rangle) = \emptyset$.

The projection of $u\langle U\rangle$ on $\mathbb{M}$ is $\mathsf{proj}_{\mathbb{M}}(u\langle U\rangle) = \bigcup_{v\langle V\rangle \in \mathbb{M}} \mathsf{proj}_{v\langle V\rangle}(u\langle U\rangle)$, and $\mathsf{proj}_{\mathbb{M}}(\mathbb{L}) = \bigcup_{u\langle U\rangle \in \mathbb{L}} \mathsf{proj}_{\mathbb{M}}(u\langle U\rangle)$.

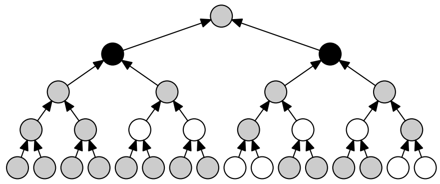
In order to grasp this definition, it might be helpful to study the example in Figure 4. Note in particular that if $u\langle U\rangle \preceq v\langle V\rangle$, then $\mathsf{proj}_{v\langle V\rangle}(u\langle U\rangle) = u\langle U\rangle$. Here and in the following, we adopt the convention that projections resulting in $\emptyset$ are implicitly eliminated from all L-configurations.

The next observation says that any L-configuration $\mathbb{L}$ can be written as a disjoint union of the sets of subconfigurations of $\mathbb{L}$ covered by each subconfiguration in $\mathsf{canon}(\mathbb{L})$, and that the cost of $\mathbb{L}$ is the sum of the costs of the sets in this disjoint union. This statement is obvious once deciphered, and the proof is immediate from Definition 7.7 and Proposition 7.5, parts 4 and 5.

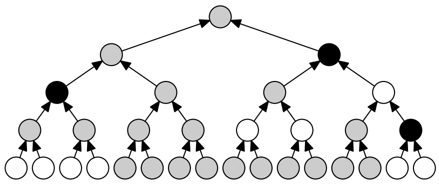
Observation 7.8. *Let* $\mathbb{L}' = \mathsf{canon}(\mathbb{L})$. *Then it holds that* $\mathbb{L}$ *is a disjoint union of the sets* $\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}) = \big\{u\langle U\rangle \mid v\langle V\rangle \succeq u\langle U\rangle \in \mathbb{L}\big\}$ *for all* $v\langle V\rangle \in \mathbb{L}'$.



(a) The L-configuration $\mathbb{L}$.



(b) The L-configuration $\mathbb{M}$.



(c) The projection $\mathsf{proj}_{\mathbb{M}}(\mathbb{L})$.

Figure 4: Example L-configurations $\mathbb{L}$ and $\mathbb{M}$ and projected L-configuration $\mathsf{proj}_{\mathbb{M}}(\mathbb{L})$.

Also, $\mathit{cost}(\mathbb{L}) = \sum_{v\langle V\rangle \in \mathbb{L}'} \mathit{cost}(\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}))$*, and for all* $v\langle V\rangle \in \mathbb{L}'$ *it holds that* $\mathit{cost}(v\langle V\rangle) \leq \mathit{cost}(\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}))$*.*

Using Proposition 7.5, Definition 7.7 and Observation 7.8, we can prove that for every overlapping L-pebbling we can find a non-overlapping pebbling which is at least as good and at least as cheap.

Lemma 7.9. *Suppose that* $\mathcal{L}$ *is an arbitrary legal L-pebbling of* T*. Then there is a non-overlapping L-pebbling* $\mathcal{L}'$ *of* T *such that* $\mathit{cost}\big(\mathcal{L}'\big) \leq \mathit{cost}\big(\mathcal{L}\big)$*.*

Proof. Given $\mathcal{L} = \big\{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\big\}$, we create the "backbone" $\mathcal{L}' = \big\{\mathbb{L}'_0, \ldots, \mathbb{L}'_\tau\big\}$ of a non-overlapping pebbling by setting $\mathbb{L}'_t = \mathsf{canon}(\mathbb{L}_t)$. By Proposition 7.5, part 5, $\mathit{cost}(\mathbb{L}'_t) \leq \mathit{cost}(\mathbb{L}_t)$, so we are done if we can fill in the holes in the transitions $\mathbb{L}'_t \rightsquigarrow \mathbb{L}'_{t+1}$ in cost $\max\big\{\mathit{cost}(\mathbb{L}_t), \mathit{cost}(\mathbb{L}_{t+1})\big\}$ using the non-overlapping moves of Definition 7.6. This is basically just an exercise in applying Proposition 7.5. Consider the moves $\mathbb{L}_t \rightsquigarrow \mathbb{L}_{t+1}$ in $\mathcal{L}$.

Introduction $\mathbb{L}_{t+1} = \mathbb{L}_t \cup v\langle \mathit{pred}(v)\rangle$: If $v\langle \mathit{pred}(v)\rangle \preceq \mathbb{L}'_t$, set $\mathbb{L}'_{t+1} = \mathbb{L}'_t$. Otherwise, introduce $v\langle \mathit{pred}(v)\rangle$ and canonize by expanding (at most three times) to get $\mathbb{L}'_{t+1} = \mathsf{canon}(\mathbb{L}_{t+1})$ in cost at most $\mathit{cost}(\mathbb{L}'_t \cup v\langle \mathit{pred}(v)\rangle) \leq \mathit{cost}(\mathbb{L}_{t+1})$ by parts 5 and 6 of Proposition 7.5. (Here we use the obvious fact that $\mathsf{canon}(v\langle \mathit{pred}(v)\rangle) = v\langle \mathit{pred}(v)\rangle$.)

Merger $\mathbb{L}_{t+1} = \mathbb{L}_t \cup \mathsf{merge}(u\langle U\rangle, v\langle V\rangle)$ for $u\langle U\rangle, v\langle V\rangle \in \mathbb{L}_t$: $\mathbb{L}_{t+1} \sim \mathbb{L}_t$, so set $\mathbb{L}'_{t+1} = \mathbb{L}'_t = \mathsf{canon}(\mathbb{L}_{t+1})$.

Reversal $\mathbb{L}_{t+1} = \mathbb{L}_t \cup v\langle V\rangle$ for $v\langle V\rangle \prec u\langle U\rangle \in \mathbb{L}_t$: $\mathbb{L}_{t+1} \sim \mathbb{L}_t$, so set $\mathbb{L}'_{t+1} = \mathbb{L}'_t$.

Erasure $\mathbb{L}_{t+1} = \mathbb{L}_t \setminus v\langle V\rangle$ for $v\langle V\rangle \in \mathbb{L}_t$: If $v\langle V\rangle \preceq \mathbb{L}_{t+1}$ we have $\mathbb{L}_{t+1} \sim \mathbb{L}_t$ and can set $\mathbb{L}'_{t+1} = \mathbb{L}'_t$, so assume that $v\langle V\rangle \npreceq \mathbb{L}_{t+1}$. Since $\mathbb{L}'_t$ is non-touching, there is a $u\langle U\rangle \in \mathbb{L}'_t$ such that $v\langle V\rangle \preceq u\langle U\rangle$. It follows from Observation 7.8 that for $w\langle W\rangle \in \mathbb{L}'_t$, $w\langle W\rangle \neq u\langle U\rangle$, we have $\mathsf{proj}_{w\langle W\rangle}(\mathbb{L}_{t+1}) = \mathsf{proj}_{w\langle W\rangle}(\mathbb{L}_t)$. Thus, letting $\mathbb{L}^u_i = \mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_i)$ for $i = t, t+1$, by Proposition 7.5, part 4, it is sufficient to show that we can implode $u\langle U\rangle = \mathsf{canon}(\mathbb{L}^u_t) = \mathsf{canon}(\mathbb{L}^u_{t+1} \cup v\langle V\rangle)$ into $\mathbb{M} = \mathsf{canon}(\mathbb{L}^u_{t+1})$ in cost at most $\max\{\textit{cost}(\mathbb{L}^u_{t+1} \cup v\langle V\rangle), \textit{cost}(\mathbb{L}^u_{t+1})\} = \textit{cost}(\mathbb{L}^u_{t+1} \cup v\langle V\rangle)$. By part 1 of the same proposition, it is enough to check that it holds that $\textit{cost}(\mathbb{M} \cup u\langle U\rangle) \leq \textit{cost}(\mathbb{L}^u_{t+1} \cup v\langle V\rangle)$. But this is just part 5 of the proposition.

Eliminating "idle moves" $\mathbb{L}'_{t+1} = \mathbb{L}'_t$, we see that we get a non-overlapping pebbling in accordance with Definition 7.6. $\square$

Lemma 7.9 tells us that as far as pebbling cost is concerned, without loss of generality we may assume that an L-pebbling $\mathcal{L}$ that reaches $z\langle\emptyset\rangle$ is non-overlapping. This completes part 1 in the proof of Lemma 6.8 sketched at the beginning of this section.

If $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ is a non-overlapping pebbling ending in an implosion $u\langle U\rangle \rightsquigarrow \mathbb{M}$, it seems natural to try to replace the moves in $\mathcal{L}$ leading to $u\langle U\rangle$ by a reversal-free pebbling reaching $\mathbb{M} \preceq u\langle U\rangle$. Since $u\langle U\rangle$ and $\mathbb{L}_\tau \setminus (u\langle U\rangle \cup \mathbb{M})$ are mutually non-touching by definition, this substitution should not affect the cost of the pebbling outside $cl(u\langle U\rangle)$. Intuitively, one natural candidate for such a substitution is the projection of $\mathcal{L}$ on $\mathbb{M}$. We next show that projecting any L-pebbling on any non-touching L-configuration $\mathbb{M}$, we get a legal L-pebbling inside $cl(\mathbb{M})$, modulo some technical details. This is part 2 in our proof outline.

Proposition 7.10. *For $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ an arbitrary L-pebbling and $\mathbb{M}$ a non-touching L-configuration, let $\mathsf{proj}_{\mathbb{M}}(\mathcal{L}) = \{\mathbb{L}'_0, \ldots, \mathbb{L}'_\tau\}$ for $\mathbb{L}'_t = \mathsf{proj}_{\mathbb{M}}(\mathbb{L}_t)$. Then $\mathsf{proj}_{\mathbb{M}}(\mathcal{L})$ is a legal L-pebbling if we eliminate idle moves $\mathbb{L}'_{t+1} = \mathbb{L}'_t$ and take care of that one reversal or erasure $\mathbb{L}_t \rightsquigarrow \mathbb{L}_{t+1}$ in $\mathcal{L}$ may correspond to a sequence of reversals or erasures respectively in $\mathsf{proj}_{\mathbb{M}}(\mathcal{L})$. Legalizing $\mathsf{proj}_{\mathbb{M}}(\mathcal{L})$ by performing these moves one by one does not affect the pebbling cost. Also, if $\mathcal{L}$ does not contain any reversals, then neither does $\mathsf{proj}_{\mathbb{M}}(\mathcal{L})$.*

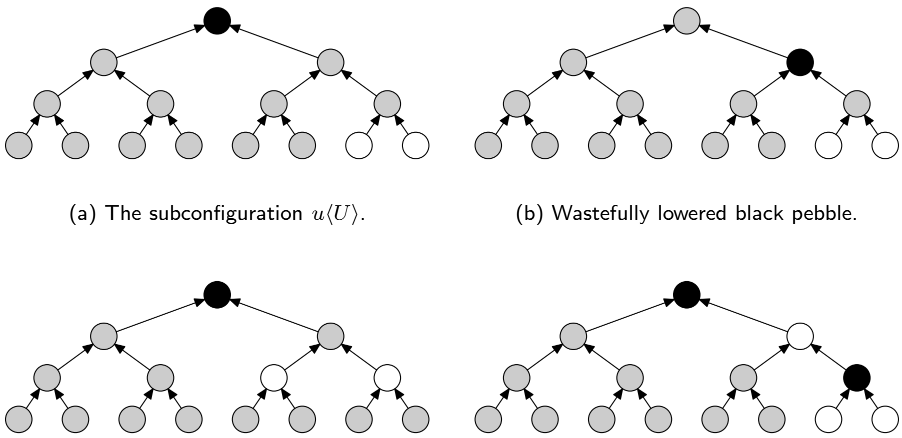
Proof. By induction over the pebbling moves $\mathbb{L}_t \rightsquigarrow \mathbb{L}_{t+1}$ in $\mathcal{L}$. Case analysis:

Introduction If $v\langle pred(v)\rangle \not\preceq \mathbb{M}$ the projection does not change, and otherwise adding $v\langle pred(v)\rangle = \mathsf{proj}_{\mathbb{M}}(v\langle pred(v)\rangle)$ is a legal introduction move.

Merger Suppose that $\mathbb{L}_{t+1} = \mathbb{L}_t \cup u\langle (U \cup V) \setminus \{v\}\rangle$ for $u\langle U\rangle, v\langle V\rangle \in \mathbb{L}_t$ such that $v \in U$. For all $w\langle W\rangle \in \mathbb{M}$ such that $v \notin int(w\langle W\rangle)$, it is straightforward, if tedious, to verify that $u\langle (U \cup V) \setminus \{v\}\rangle$ projects the same subconfigurations on $w\langle W\rangle$ as do $u\langle U\rangle$ and $v\langle V\rangle$ together. Suppose that $v \in int(w\langle W\rangle)$. Since $\mathbb{M}$ is non-touching there is at most one such $w\langle W\rangle \in \mathbb{M}$, and $\mathsf{proj}_{w\langle W\rangle}(u\langle (U \cup V) \setminus \{v\}\rangle)$ can be verified to be a legal merger of $\mathsf{proj}_{w\langle W\rangle}(u\langle U\rangle)$ and $\mathsf{proj}_{w\langle W\rangle}(v\langle V\rangle)$.

Reversal If $v\langle V\rangle \preceq u\langle U\rangle$ it holds that $\mathsf{proj}_{\mathbb{M}}(v\langle V\rangle) \preceq \mathsf{proj}_{\mathbb{M}}(u\langle U\rangle)$, so adding $\mathsf{proj}_{\mathbb{M}}(v\langle V\rangle)$ is a sequence of legal reversals. As this sequence of reversals is performed, the pebbling cost increases monotonously by part 1 of Proposition 7.5.

Erasure If $\mathbb{L}_{t+1} = \mathbb{L}_t \setminus v\langle V\rangle$ for $v\langle V\rangle \in \mathbb{L}_t$, removing $\mathsf{proj}_{\mathbb{M}}(v\langle V\rangle)$ from $\mathbb{L}'_t$ is a sequence of legal erasures. As this sequence of erasures is performed, the pebbling cost decreases monotonously by part 1 of Proposition 7.5.



(a) The subconfiguration $u\langle U\rangle$.

(b) Wastefully lowered black pebble.

(c) Superfluous white pebble.

(d) Wasteful "split" of $u\langle U\rangle$.

Figure 5: A subconfiguration $u\langle U\rangle$ and three wasteful implosions of $u\langle U\rangle$.

We see that the cost of this pebbling is $\max_{t\in[\tau]}\{\mathsf{proj}_{\mathbb{M}}(\mathbb{L}_t)\}$, and if $\mathcal{L}$ is reversal-free then so is $\mathsf{proj}_{\mathbb{M}}(\mathcal{L})$, since every move in $\mathcal{L}$ is matched by the same kind of moves in $\mathsf{proj}_{\mathbb{M}}(\mathcal{L})$. □

In view of Proposition 7.10, the transformation from a non-overlapping pebbling $\mathcal{L}$ to a reversal-free pebbling $\mathcal{L}'$ seems obvious: by forward induction over the moves in $\mathcal{L}$, replace each implosion $u\langle U\rangle \rightsquigarrow \mathbb{M}$ at time t by a local projection of $\{\mathbb{L}_0,\ldots,\mathbb{L}_t\}$ on $\mathbb{M}$. Since by induction there are no reversals before time t, the projection must be a reversal-free pebbling inside $cl(\mathbb{M})$. Doing this for all implosions, we get a globally reversal-free pebbling $\mathcal{L}'$ ending in $z\langle\emptyset\rangle$. This is the transformation described in part 3 of our road map for the proof of Lemma 6.8.

There is only one problem. Although $\mathsf{proj}_{\mathbb{M}}(\mathcal{L})$ is a legal L-pebbling, it is not true in general that $\mathit{cost}(\mathsf{proj}_{\mathbb{M}}(\mathcal{L})) \leq \mathit{cost}(\mathcal{L})$. For instance, if $v\langle V\rangle \preceq u\langle\emptyset\rangle$ for $V \neq \emptyset$, then $\mathsf{proj}_{v\langle V\rangle}(u\langle\emptyset\rangle) = v\langle V\rangle$ and hence $\mathit{cost}(\mathsf{proj}_{v\langle V\rangle}(u\langle\emptyset\rangle)) = 1 + |V| > \mathit{cost}(u\langle\emptyset\rangle) = 1$. Looking at this counterexample, however, it seems clear that having gotten as far as $u\langle\emptyset\rangle$, reversing to the weaker and more expensive configuration $v\langle V\rangle$ is non-optimal. What we want to do next is to define formally which reversals are *wasteful* in this sense, and to prove that for pebblings avoiding such wasteful reversals, projection does not increase the pebbling cost.

Since the definition of wastefulness turns out to be quite technical, we first try to give some more intuition for which kind of reversals we disapprove of.

Example 7.11. Consider the subconfiguration $u\langle U\rangle$ in Figure 5(a).

1. If $v \in T^u_*$, the reversal $u\langle U\rangle \rightsquigarrow v\langle T^v_* \cap U\rangle$ is acceptable only if $T^v_* \cap U \neq U$, i.e., if we get rid of white pebbles by lowering the black pebble from u to v. The reversal in Figure 5(b) does not satisfy this.

2. For V a simple roof below u over U, we approve of $u\langle U\rangle \rightsquigarrow u\langle V\rangle$ only if for all $w \in V$ it holds that $T^w \cap U \neq \emptyset$. Otherwise, unnecessary white pebbles have been introduced, as in Figure 5(c).

3. If $u\langle U\rangle$ is imploded into non-touching $\{v_1\langle V_1\rangle, v_2\langle V_2\rangle\}$ such that, say, $v_2 \in T_*^{v_1}$, it should not be the case that $v_1\big\langle\big(V_1 \setminus P^{v_2}\big) \cup V_2\big\rangle \preceq u\langle U\rangle$, for if so we could have reversed to this stronger subconfiguration instead of $\{v_1\langle V_1\rangle, v_2\langle V_2\rangle\}$ at no extra cost. The implosion in Figure 5(d) violates this condition.

The reversals from $u\langle U\rangle$ in figures 5(b), 5(c) and 5(d) are all examples of wasteful implosions for which our reversal-free pebbling $\mathcal{L}'$ constructed by projection may become more expensive than $\mathcal{L}$. Looking at these examples, it is easy to believe that such moves are non-optimal and that it ought to be possible to eliminate them. The formal definition of wastefulness is as follows.

Definition 7.12 (Wasteful implosion). For a non-touching L-configuration $\mathbb{M} = \{v_i\langle V_i\rangle \mid i \in [m]\} \preceq u\langle U\rangle$, the implosion $u\langle U\rangle \rightsquigarrow \mathbb{M}$ is *non-wasteful* if

1. for every $v \in Bl\big(\mathbb{M}\big) \setminus \{u\}$ there is a $w \in U \cap T_*^{succ(v)}$ such that it holds for the path $p_v = P^w \setminus P_*^{succ(v)}$ that $p_v \cap \big(Bl\big(\mathbb{M}\big) \cup Wh\big(\mathbb{M}\big)\big) = \emptyset$,

2. for every $v \in Wh\big(\mathbb{M}\big)$ there is a $w \in U \cap T^v$ such that it holds for the path $p_v = P^w \setminus P_*^v$ that $p_v \cap Bl\big(\mathbb{M}\big) = \emptyset$,

3. the paths above from $\big(Bl\big(\mathbb{M}\big) \cup Wh\big(\mathbb{M}\big)\big) \setminus \{u\}$ to $Wh\big(u\langle U\rangle\big) = U$ can all be chosen pairwise disjoint, i.e., such that $p_v \cap p_{v'} = \emptyset$ if $v \neq v'$.

If $u\langle U\rangle \rightsquigarrow \mathbb{M}$ is not a non-wasteful implosion it is said to be *wasteful*.

Definition 7.12 identifies the offending reversal moves for which our projective construction of a reversal-free but cheap pebbling fails. Continuing according to part 4 in our proof plan, we show that for pebblings without such wasteful moves the projective construction works. This is the next lemma.

Lemma 7.13. *Suppose that* $\mathcal{L} = \big\{\mathbb{L}_0, \ldots, \mathbb{L}_{\tau-2}, \mathbb{L}_{\tau-1} = u\langle U\rangle \rightsquigarrow \mathbb{M}\big\}$ *is a pebbling without reversals except for a final non-wasteful implosion* $u\langle U\rangle \rightsquigarrow \mathbb{M}$*. Then* $\mathit{cost}(\mathsf{proj}_{\mathbb{M}}(\mathbb{L}_t)) \leq \mathit{cost}(\mathbb{L}_t)$ *for all* $t < \tau$*, and* $\mathit{cost}(\mathsf{proj}_{\mathbb{M}}(\mathcal{L})) \leq \mathit{cost}(\mathcal{L})$*.*

Proof. Let $\mathbb{L}'_t = \mathsf{proj}_{\mathbb{M}}(\mathbb{L}_t)$ for all $t < \tau$. By Proposition 7.10, it suffices to show $\mathit{cost}(\mathbb{L}'_t) \leq \mathit{cost}(\mathbb{L}_t)$ to get $\mathit{cost}(\mathsf{proj}_{\mathbb{M}}(\mathcal{L})) \leq \mathit{cost}(\mathcal{L})$. We observe that by the proof of Lemma 6.9 on page 19, $\mathit{cover}(\mathbb{L}_t)$ grows monotonously with t in a non-redundant reversal-free pebbling, so in particular $\mathbb{L}_t \preceq u\langle U\rangle$ for all t.

Clearly, to prove $\mathit{cost}(\mathbb{L}'_t) \leq \mathit{cost}(\mathbb{L}_t)$ it is enough to find for each vertex $v \in Bl\big(\mathbb{L}'_t\big) \cup Wh\big(\mathbb{L}'_t\big)$ an associated vertex $v_L \in Bl\big(\mathbb{L}_t\big) \cup Wh\big(\mathbb{L}_t\big)$ such that $v_L \neq w_L$ if $v \neq w$. If $v \in \big(Bl\big(\mathbb{L}'_t\big) \cup Wh\big(\mathbb{L}'_t\big)\big) \cap \big(Bl\big(\mathbb{L}_t\big) \cup Wh\big(\mathbb{L}_t\big)\big)$, an obvious choice is $v_L = v$. Suppose therefore that $v \in \big(Bl\big(\mathbb{L}'_t\big) \cup Wh\big(\mathbb{L}'_t\big)\big) \setminus \big(Bl\big(\mathbb{L}_t\big) \cup Wh\big(\mathbb{L}_t\big)\big)$. Then $v \in \partial\mathbb{M}$, since it is easy to check that $v \in \mathit{int}(\mathbb{M})$ implies $v \in Bl\big(\mathbb{L}_t\big) \cup Wh\big(\mathbb{L}_t\big)$. Also, there is a subconfiguration $w_v\big\langle W_v\big\rangle \in \mathbb{L}_t$ such that $v \in \mathit{int}\big(w_v\big\langle W_v\big\rangle\big)$, namely the $w_v\big\langle W_v\big\rangle$ projecting the pebble on v. Lastly, note that if $v \in \big(Bl\big(\mathbb{L}'_t\big) \cup Wh\big(\mathbb{L}'_t\big)\big) \cap \partial\mathbb{M}$, it is a routine matter to verify that v has the same colour in $\mathbb{L}'_t$ and $\mathbb{M}$, i.e., either $v \in Bl\big(\mathbb{L}'_t\big) \cap Bl\big(\mathbb{M}\big)$ or $v \in Wh\big(\mathbb{L}'_t\big) \cap Wh\big(\mathbb{M}\big)$. We choose $v_L \in Bl\big(\mathbb{L}_t\big) \cup Wh\big(\mathbb{L}_t\big)$ for such vertices v by first associating a unique $v_u \in U = Wh\big(u\langle U\rangle\big)$ to v as follows.

1. If $v \in Bl(\mathbb{L}'_t) \cap Bl(\mathbb{M})$, pick a vertex $v_u \in \left(U \cap T_*^{succ(v)}\right) \setminus T^v$ and a path $p_v = P^{v_u} \setminus P_*^{succ(v)}$ to v_u such that $p_v \cap \left(Bl(\mathbb{M}) \cup Wh(\mathbb{M})\right) = \emptyset$ as guaranteed by Definition 7.12. For the subconfiguration $w_v\langle W_v\rangle \in \mathbb{L}_t$ projecting the black pebble on v, we must have $succ(v) \in cover\left(w_v\langle W_v\rangle\right)$ since $v \in int\left(w_v\langle W_v\rangle\right)$, and consequently $succ(v) \in p_v \cap cover\left(w_v\langle W_v\rangle\right) \neq \emptyset$.

2. If $v \in Wh(\mathbb{L}'_t) \cap Wh(\mathbb{M})$, pick $v_u \in U \cap T^v$ and $p_v = P^{v_u} \setminus P_*^v$ such that $p_v \cap Bl(\mathbb{M}) = \emptyset$ as guaranteed by the definition. For $w_v\langle W_v\rangle \in \mathbb{L}_t$ projecting the white pebble on v, we have $v \in int\left(w_v\langle W_v\rangle\right) \subseteq cover\left(w_v\langle W_v\rangle\right)$, so $v \in p_v \cap cover\left(w_v\langle W_v\rangle\right) \neq \emptyset$.

According to Definition 7.12 all the paths p_v above can be chosen disjoint.

We now use these paths to choose a distinct $v_L \in Bl(\mathbb{L}_t) \cup Wh(\mathbb{L}_t)$ for each $v \in \left(Bl(\mathbb{L}'_t) \cup Wh(\mathbb{L}'_t)\right) \setminus \left(Bl(\mathbb{L}_t) \cup Wh(\mathbb{L}_t)\right)$. By construction, $\mathbb{L}'_t \preceq \mathbb{L}_t \preceq u\langle U\rangle$, and in particular $w_v\langle W_v\rangle \preceq u\langle U\rangle$ for all $w_v\langle W_v\rangle$ found above. Note that $p_v \cap cover\left(w_v\langle W_v\rangle\right) \neq \emptyset$ and that $p_v \not\subseteq cover(u\langle U\rangle)$ since the lowest vertex in p_v is a white pebble of $u\langle U\rangle$. This implies that $W_v \cap p_v \neq \emptyset$, for otherwise $p_v \subseteq cover(w_v\langle W_v\rangle)$ which yields the contradiction $w_v\langle W_v\rangle \not\preceq u\langle U\rangle$. Thus we can choose $v_L \in Bl(\mathbb{L}_t) \cup Wh(\mathbb{L}_t)$ to be the vertex in $W_v \cap p_v$.

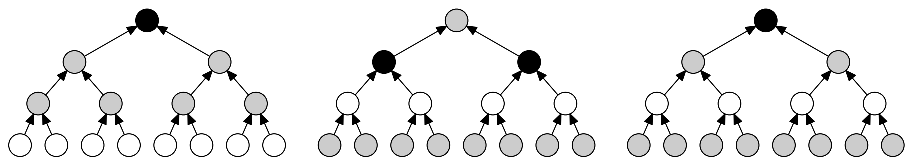
Since all paths p_v are disjoint, all v_L chosen as just described are distinct. They must also be distinct from all $v \in \left(Bl(\mathbb{L}'_t) \cup Wh(\mathbb{L}'_t)\right) \cap \left(Bl(\mathbb{L}_t) \cup Wh(\mathbb{L}_t)\right)$. To see this, first observe that for all paths p_v in Definition 7.12 in holds that $\left(p_v \setminus \{v\}\right) \cap \left(Bl(\mathbb{M}) \cup Wh(\mathbb{M})\right) = \emptyset$ (for paths from vertices in $Wh(\mathbb{M})$ this follows from the fact that $\mathbb{M}$ is non-touching). Again, by definition, the topmost vertex in $p_v \setminus \{v\}$ (if $p_v \neq \{v\}$) is outside $cl(\mathbb{M})$, and combining these two observations we get that $\left(p_v \setminus \{v\}\right) \cap cl(\mathbb{M}) = \emptyset$. In particular, for our chosen $v_L \in p_v \setminus \{v\}$ it holds that $v_L \notin cl(\mathbb{M})$, and thus $v_L \notin Bl(\mathbb{L}'_t) \cup Wh(\mathbb{L}'_t)$ for $\mathbb{L}'_t = \mathsf{proj}_{\mathbb{M}}(\mathbb{L}_t)$, since all projected pebbles must lie in $cl(\mathbb{M}) \cap cl(\mathbb{L}_t)$ by definition. Hence, all associated vertices v_L are distinct, so $\mathsf{cost}(\mathbb{L}'_t) \leq \mathsf{cost}(\mathbb{L}_t)$. □

We can use Lemma 7.13 to eliminate non-wasteful implosions one by one. If $\mathcal{L} = \left\{\mathbb{L}_0, \ldots, \mathbb{L}_\tau \rightsquigarrow (\mathbb{L}_\tau \setminus u\langle U\rangle) \cup \mathbb{M}\right\}$ is a non-overlapping reversal-free pebbling except for a final non-wasteful implosion $u\langle U\rangle \rightsquigarrow \mathbb{M}$, then by definition $\mathbb{L}_\tau = \{v_i\langle V_i\rangle \mid i \in [n]\}$ is non-touching, and using Observation 7.8 each $\mathbb{L}_t$ can be written as a non-touching union of $\mathbb{L}_t^{v_i} = \mathsf{proj}_{v_i\langle V_i\rangle}(\mathbb{L}_t)$ such that $\mathsf{cost}(\mathbb{L}_t) = \sum_{v_i\langle V_i\rangle \in \mathbb{L}_\tau} \mathsf{cost}(\mathbb{L}_t^{v_i})$. For all $v_i\langle V_i\rangle \in \mathbb{L}_\tau$, $\mathcal{L}^{v_i} = \left\{\mathbb{L}_0^{v_i}, \ldots, \mathbb{L}_{\tau-1}^{v_i}, \{v_i\langle V_i\rangle\}\right\}$ can be seen to be pairwise non-touching pebblings without reversals such that $\left\{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\right\}$ is the union of all local pebblings $\mathbb{L}_t^{v_i}$. It follows by Lemma 7.13 (and Proposition 7.5, part 4) that we can locally replace $\mathcal{L}^u$ for the imploded subconfiguration $u\langle U\rangle$ by $\mathsf{proj}_{\mathbb{M}}(\mathcal{L}^u)$ in $\mathcal{L}$ without increasing the global pebbling cost. Doing this by forward induction for all implosions in turn, we get the following corollary.

Corollary 7.14. *Let $\mathcal{L} = \left\{\mathbb{L}_0, \ldots, \mathbb{L}_\tau = \{z\langle\emptyset\rangle\}\right\}$ be a non-overlapping L-pebbling of T without wasteful implosions. Then there is an L-pebbling $\mathcal{L}'$ of T without any reversal moves such that $\mathsf{cost}(\mathcal{L}') \leq \mathsf{cost}(\mathcal{L})$.*

This concludes part 4 in the proof outline on page 21.

All that remains is to show that in an arbitrary non-overlapping L-pebbling we can always replace wasteful implosions by non-wasteful ones without increasing the pebbling cost by more than a constant factor. It will take a couple



(a) The subconfiguration. (b) Wasteful implosion $\mathbb{L}$. (c) Non-wasteful $\mathbb{M} \succ \mathbb{L}$.

Figure 6: Illustration of case 1 in the proof of Lemma 7.15.

of technical lemmas before we get there, but the intuition from Example 7.11 is clear: if $\mathbb{L}_t \rightsquigarrow \mathbb{L}_{t+m+1}$ is a wasteful implosion, we should be able to match this move with a non-wasteful implosion $\mathbb{L}'_t \rightsquigarrow \mathbb{L}'_{t+m+1}$ instead, where $\mathbb{L}'_i \succeq \mathbb{L}_i$ and $\mathsf{cost}(\mathbb{L}'_i) \leq \mathsf{cost}(\mathbb{L}_i)$ for $i = t, t+m+1$. The only thing that complicates the matter is that we may have to pay extra for the transitional L-configurations during the implosion $\mathbb{L}'_t \rightsquigarrow \mathbb{L}'_{t+m+1}$ because of overlapping subconfigurations.

The cornerstone of our proof is the fact that for every wasteful implosion $u\langle U\rangle \rightsquigarrow \mathbb{L}$, there is a non-wasteful implosion to $\mathbb{M} \succ \mathbb{L}$ with $\mathsf{cost}(\mathbb{M}) \leq \mathsf{cost}(\mathbb{L})$.

Lemma 7.15. *If* $u\langle U\rangle \rightsquigarrow \mathbb{L}$ *is a wasteful implosion, then there is a non-touching* $\mathbb{M}$ *such that* $u\langle U\rangle \succeq \mathbb{M} \succ \mathbb{L}$*,* $\mathsf{cost}(\mathbb{M}) \leq \min\{\mathsf{cost}(u\langle U\rangle), \mathsf{cost}(\mathbb{L})\}$ *and* $u\langle U\rangle \rightsquigarrow \mathbb{M}$ *is a non-wasteful implosion.*

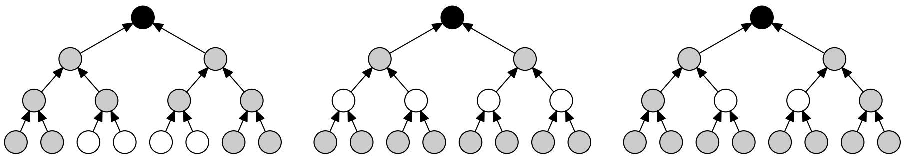
Proof. If $u\langle U\rangle \rightsquigarrow \mathbb{M}$ is a non-wasteful implosion, it holds that $\mathsf{cost}(\mathbb{M}) = \left|Bl(\mathbb{M})\right| + \left|Wh(\mathbb{M})\right| \leq \mathsf{cost}(u\langle U\rangle) = 1 + |U|$, since by Definition 7.12 every $v \in \left(Bl(\mathbb{M}) \cup Wh(\mathbb{M})\right) \setminus \{u\}$ can be associated with a distinct $w \in U$.

We demonstrate that if $u\langle U\rangle \rightsquigarrow \mathbb{L}$ is a wasteful implosion, we can find an $\mathbb{M}$ such that $u\langle U\rangle \succeq \mathbb{M} \succ \mathbb{L}$ and $\mathsf{cost}(\mathbb{M}) \leq \mathsf{cost}(\mathbb{L})$. If $u\langle U\rangle \rightsquigarrow \mathbb{M}$ is also a wasteful implosion, we repeat this construction. Sooner or later the process must terminate for some $\mathbb{M} \preceq u\langle U\rangle$ such that $u\langle U\rangle \rightsquigarrow \mathbb{M}$ is non-wasteful—if nothing else, by definition the trivial implosion $u\langle U\rangle \rightsquigarrow u\langle U\rangle$ is.

According to Definition 7.12, the configuration $\mathbb{L} = \left\{v_i\langle V_i\rangle\right\}$ can be wasteful with respect to $u\langle U\rangle$ in three ways.

1. Some black pebble $v \in Bl(\mathbb{L}) \setminus \{u\}$ lacks a path. If $succ(v) \in Wh(\mathbb{L})$ we must have $succ(v) \in cover(u\langle U\rangle)$, so we can add $\mathsf{canon}(\{succ(v)\}) = succ(v)\langle v, sibl(v)\rangle$ to $\mathbb{L}$ and set $\mathbb{M} = \mathsf{canon}(\mathbb{L} \cup succ(v)\langle v, sibl(v)\rangle) \succ \mathbb{L}$ with $\mathsf{cost}(\mathbb{M}) \leq \mathsf{cost}(\mathbb{L}) + |\{sibl(v)\}| - |\{v, succ(v)\}| < \mathsf{cost}(\mathbb{L})$. Otherwise, all paths from $succ(v)$ downwards in $T^{sibl(v)}$ are either blocked by $r_1, \ldots, r_m \in Bl(\mathbb{L}) \cap T^{sibl(v)}$ or reach sources in $T^{sibl(v)}$ without passing pebbled vertices (we can have $m = 0$). From this we can conclude that $V = T^{succ(v)} \setminus \left(T^v \cup \bigcup_{i\in[m]} T^{r_i}\right) \subseteq cover(u\langle U\rangle)$, so we can add $\mathsf{canon}(V) = succ(v)\langle v, r_1, \ldots, r_m\rangle \preceq u\langle U\rangle$ to $\mathbb{L}$, which increases the cost by 1 for $succ(v)$. Setting $\mathbb{M} = \mathsf{canon}(\mathbb{L} \cup succ(v)\langle v, r_1, \ldots, r_m\rangle) \succ \mathbb{L}$ removes the pebbles from v and $r_1, \ldots, r_m$ and decreases the cost by at least 1, so $\mathsf{cost}(\mathbb{M}) \leq \mathsf{cost}(\mathbb{L})$. See Figure 6 for a simple example.

2. There is a white pebble $w \in Wh(\mathbb{L})$ such that all paths downwards in T^w either are blocked by $r_1, \ldots, r_m \in Bl(\mathbb{L}) \cap T^w_*$ or reach sources in T^w



(a) The subconfiguration. (b) Wasteful implosion $\mathbb{L}$. (c) Non-wasteful $\mathbb{M} \succ \mathbb{L}$.

Figure 7: Illustration of case 2 in the proof of Lemma 7.15.

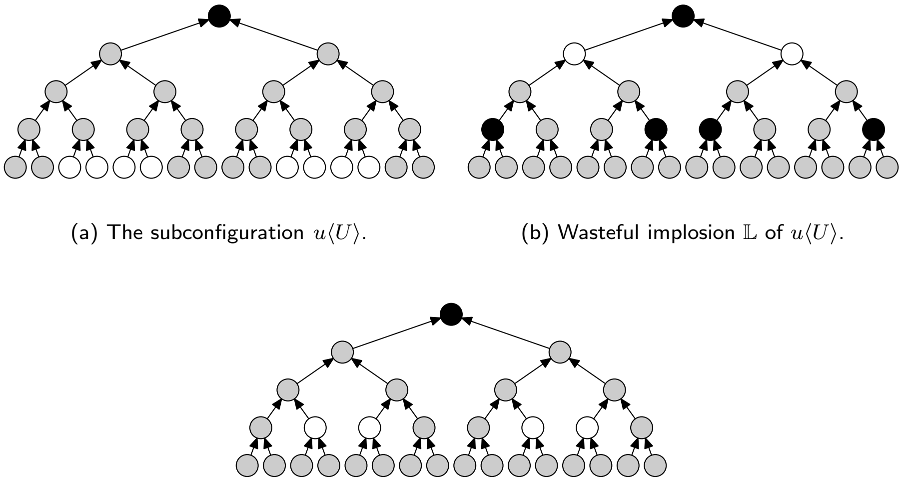
without passing pebbled vertices. If so, we have $V = T^w \setminus \bigcup_{i\in[m]} T^{r_i} \subseteq$ $cover(u\langle U\rangle)$, and we can add $\mathsf{canon}(V) = w\langle r_1, \ldots, r_m\rangle \preceq u\langle U\rangle$ to $\mathbb{L}$ at no extra cost and set $\mathbb{M} = \mathsf{canon}(\mathbb{L} \cup w\langle r_1, \ldots, r_m\rangle) \succ \mathbb{L}$. Here we get a strict inequality $\textit{cost}(\mathbb{M}) < \textit{cost}(\mathbb{L})$ since the canonization eliminates at least the pebbles on w. This case is illustrated in Figure 7.

3. There are paths for all $v \in \big(Bl(\mathbb{L}) \cup Wh(\mathbb{L})\big) \setminus \{u\}$ to $w \in U$, but they cannot be chosen disjoint. Start picking disjoint paths bottom-up so that when we choose a path for a white pebble $v \in Wh(\mathbb{L})$ we have already determined paths for all $w \in \big(Bl(\mathbb{L}) \cup Wh(\mathbb{L})\big) \cap T^v_*$, and when we choose a path for a black pebble $v \in Bl(\mathbb{L})$ we have already determined paths for all $w \in \big(Bl(\mathbb{L}) \cup Wh(\mathbb{L})\big) \cap T^{sibl(v)}_*$, i.e., for all of $T^{succ(v)} \setminus \{v\}$. For note that for black pebbles, the vertex $sibl(v)$ itself cannot be black-pebbled in $\mathbb{L}$, for if so there would be no path for v and we would have case 1. For the same reason, $succ(v)$ is not white-pebbled in $\mathbb{L}$, and then $sibl(v)$ cannot be white-pebbled or $succ(v)$ black-pebbled either since $\mathbb{L}$ is non-touching.

   At some point we reach a v such that no matter how we choose the paths below, we cannot choose a disjoint path for v. Consider the colour of v.

   (a) v is black. There are white pebbles in $U \cap T^{sibl(v)}_*$ reachable from v, but they are all blocked by paths already chosen from $r_1, \ldots, r_m \in Bl(\mathbb{L}) \cap T^{sibl(v)}_*$. This means that $\{succ(r_i) \mid i \in [m]\} \subseteq cover(u\langle U\rangle)$, so we can add the subconfigurations $\mathsf{canon}(\{succ(r_i) \mid i \in [m]\}) = \big\{succ(r_i)\langle r_i, sibl(r_i)\rangle \mid i \in [m]\big\}$ to $\mathbb{L}$ at an additional cost $2m$. Reasoning in the same way, we can also include the subconfiguration $succ(v)\langle v, succ(r_1), \ldots, succ(r_m)\rangle$ at a further cost of 1 for the unpebbled vertex $succ(v)$. When we canonize, the pebbles on the vertices $v, r_1, \ldots, r_m, succ(r_1), \ldots, succ(r_m)$ all disappear and the cost decreases by $2m + 1$, resulting in $\mathbb{M} \succ \mathbb{L}$ with $\textit{cost}(\mathbb{M}) \leq \textit{cost}(\mathbb{L})$.

   (b) v is white. The construction is analogous. Let the blocking black pebbles be $r_1, \ldots, r_m \in Bl(\mathbb{L}) \cap T^v_*$. Again $succ(r_i)\langle r_i, sibl(r_i)\rangle$, $i \in [m]$, can be added at an extra cost $2m$. Since $succ(r_i)$, $i \in [m]$, block all paths from v we have $T^v \setminus \bigcup_{i\in[m]} T^{succ(r_i)} \subseteq cover(u\langle U\rangle)$, so $v\langle succ(r_1), \ldots, succ(r_m)\rangle$ can be added as well at no additional cost. Canonizing decreases the cost by $2m + 1$, which yields $\mathbb{M} \succ \mathbb{L}$ with $\textit{cost}(\mathbb{M}) < \textit{cost}(\mathbb{L})$. The transition from Figure 8(b) to Figure 8(c) is accomplished by applying this construction twice.



(a) The subconfiguration $u\langle U\rangle$.

(b) Wasteful implosion $\mathbb{L}$ of $u\langle U\rangle$.

(c) Non-wasteful implosion $u\langle U\rangle \rightsquigarrow \mathbb{M} \succ \mathbb{L}$.

Figure 8: Illustration of case 3 in the proof of Lemma 7.15.

In all cases we can find a non-touching L-configuration $\mathbb{M}$ such that $u\langle U\rangle \succeq \mathbb{M} \succ \mathbb{L}$ and $\mathit{cost}(\mathbb{M}) \leq \mathit{cost}(\mathbb{L})$. The lemma follows. $\square$

The following transitivity property of non-wasteful implosions is immediate from Definition 7.12.

Observation 7.16. *If $u\langle U\rangle \rightsquigarrow \{v_i\langle V_i\rangle \mid i \in [m]\}$ and $v_i\langle V_i\rangle \rightsquigarrow \mathbb{M}_i$ for $i \in [m]$ are all non-wasteful implosions, then $u\langle U\rangle \rightsquigarrow \{\mathbb{M}_i \mid i \in [m]\}$ is a non-wasteful implosion.*

Proof. Just concatenate the paths from $\big(Bl\big(\mathbb{M}_i\big) \cup Wh\big(\mathbb{M}_i\big)\big) \setminus \{v_i\}$ to V_i with the paths from $\big(Bl\big(v_i\langle V_i\rangle\big) \cup Wh\big(v_i\langle V_i\rangle\big)\big) \setminus \{u\} = (\{v_i\} \cup V_i) \setminus \{u\}$ to U for all $i \in [m]$. $\square$

It follows from Observation 7.16, that if $u\langle U\rangle \rightsquigarrow \mathbb{L}$ is a wasteful implosion and $u\langle U\rangle \rightsquigarrow \mathbb{M} \succ \mathbb{L}$ is a corresponding non-wasteful implosion for $\mathbb{M}$ minimal, then all nontrivial "local implosions" from subconfigurations in $\mathbb{M}$ to sets of subconfigurations in $\mathbb{L}$ are wasteful. We formalize this as a lemma.

Lemma 7.17. *Suppose that $u\langle U\rangle \rightsquigarrow \mathbb{L}$ is a wasteful implosion and let $\mathbb{M} \succ \mathbb{L}$ be minimal such that $u\langle U\rangle \rightsquigarrow \mathbb{M}$ is non-wasteful. Then for each $v\langle V\rangle \in \mathbb{M}$ and each non-touching $\mathbb{L}'$ such that $\mathbb{M} \succ \mathbb{L}' \succeq \mathbb{L}$, either $\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}') = v\langle V\rangle$ or $v\langle V\rangle \rightsquigarrow \mathsf{proj}_{v\langle V\rangle}(\mathbb{L}')$ is a wasteful implosion. In particular, for each $v\langle V\rangle \in \mathbb{M}$ it holds that $\mathit{cost}(v\langle V\rangle) \leq \mathit{cost}(\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}'))$.*

Proof. Suppose that there are $v\langle V\rangle \in \mathbb{M}$ and $\mathbb{L}'$ such that $\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}') \prec v\langle V\rangle$ and $v\langle V\rangle \rightsquigarrow \mathsf{proj}_{v\langle V\rangle}(\mathbb{L}')$ is a non-wasteful implosion. Then by the transitivity in Observation 7.16 it holds that $\mathbb{M}' = \big(\mathbb{M} \cup \mathsf{proj}_{v\langle V\rangle}(\mathbb{L}')\big) \setminus v\langle V\rangle \prec \mathbb{M}$ is a non-wasteful implosion of $u\langle U\rangle$. This contradicts the minimality of $\mathbb{M}$.

If $v\langle V\rangle \rightsquigarrow \mathsf{proj}_{v\langle V\rangle}(\mathbb{L}')$ is a wasteful implosion, Lemma 7.15 says that there is a non-wasteful implosion to an L-configuration $\mathbb{M}' \succ \mathsf{proj}_{v\langle V\rangle}(\mathbb{L}')$ such that $\mathit{cost}(\mathbb{M}') \leq \mathit{cost}(\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}'))$. But we have just proven that this non-wasteful $\mathbb{M}'$ must be identical with $v\langle V\rangle$, so $\mathit{cost}(v\langle V\rangle) \leq \mathit{cost}(\mathsf{proj}_{v\langle V\rangle}(\mathbb{L}'))$. $\square$

Very roughly, the next lemma says that wasteful implosions are preserved under mergers.

Lemma 7.18. *Suppose for $i = 1, 2$ that $u_i\langle U_i\rangle \succeq \mathbb{L}_i$ and $\mathit{cost}(u_i\langle U_i\rangle) \leq \mathit{cost}(\mathbb{L}_i)$ for $\mathbb{L}_i$ non-overlapping, and that $u_1\langle U_1\rangle$ and $u_2\langle U_2\rangle$ are mutually non-overlapping with $u_2 \in U_1$. Then $\mathit{cost}(\mathsf{merge}(u_1\langle U_1\rangle, u_2\langle U_2\rangle)) \leq \mathit{cost}(\mathbb{L}_1 \cup \mathbb{L}_2)$.*

Proof. The L-configurations $\mathbb{L}_1$ and $\mathbb{L}_2$ must be mutually non-overlapping since they are covered by $u_1\langle U_1\rangle$ and $u_2\langle U_2\rangle$, respectively. Now the only way that $\mathit{cost}(\mathbb{L}_1 \cup \mathbb{L}_2)$ could be less than $\mathit{cost}(\mathsf{merge}(u_1\langle U_1\rangle, u_2\langle U_2\rangle)) = \mathit{cost}(u_1\langle U_1\rangle) + \mathit{cost}(u_2\langle U_2\rangle) - 1 \leq \mathit{cost}(\mathbb{L}_1) + \mathit{cost}(\mathbb{L}_2) - 1$ is if there were at least two vertices in $\bigcap_{i=1,2}\big(\mathit{Bl}\big(\mathbb{L}_i\big) \cup \mathit{Wh}\big(\mathbb{L}_i\big)\big)$. But $\mathit{Bl}\big(\mathbb{L}_i\big) \cup \mathit{Wh}\big(\mathbb{L}_i\big) \subseteq \mathit{cl}(\mathbb{L}_i) \subseteq \mathit{cl}(u_i\langle U_i\rangle)$ since $\mathbb{L}_i \preceq u_i\langle U_i\rangle$ by the assumptions of the lemma, and also by assumption $\mathit{cl}(u_1\langle U_1\rangle) \cap \mathit{cl}(u_1\langle U_1\rangle) = \{u_2\}$, so this is impossible. $\square$

Combining Lemmas 7.17 and 7.18, we can provide the fifth and final component in the proof of Lemma 6.8, namely that any non-overlapping L-pebbling $\mathcal{L}$ can be transformed into a pebbling $\mathcal{L}'$ without wasteful implosions such that $\mathcal{L}'$ has asymptotically the same cost as $\mathcal{L}$.

Lemma 7.19. *Suppose that $\mathcal{L}$ is a non-overlapping L-pebbling of T. Then there is a non-overlapping pebbling $\mathcal{L}'$ of T without wasteful implosions such that $\mathit{cost}(\mathcal{L}') \leq 2 \cdot \mathit{cost}(\mathcal{L})$.*

Proof. Given a non-overlapping L-pebbling $\mathcal{L}$, we build a non-overlapping pebbling $\mathcal{L}'$ without wasteful implosions such that if we let $\mathbb{L}_i \in \mathcal{L}$ denote the starting configuration of the ith move according to the rules in Definition 7.6, there is a corresponding $\mathbb{L}'_i \in \mathcal{L}'$ such that the following invariants hold:

1. $\mathbb{L}'_i$ is non-touching.
2. $\mathbb{L}'_i \succeq \mathbb{L}_i$.
3. For all $u\langle U\rangle \in \mathbb{L}'_i$, it holds that $\mathit{cost}(u\langle U\rangle) \leq \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_i))$.
4. The cost of the L-pebbling transition from $\mathbb{L}'_{i-1}$ to $\mathbb{L}'_i$ in $\mathcal{L}'$ does not exceed $2 \cdot \max\big\{\mathit{cost}(\mathbb{L}_{i-1}), \mathit{cost}(\mathbb{L}_i)\big\}$.

To see that the lemma follows from this, note that invariants 1 and 2 imply that for every $v\langle V\rangle \in \mathbb{L}_i$ there is a $u\langle U\rangle \in \mathbb{L}'_i$ with $u\langle U\rangle \succeq v\langle V\rangle$. Plugging invariant 3 into Proposition 7.5, part 4, and using that $\bigcup_{u\langle U\rangle \in \mathbb{L}'_i} \mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_i) = \mathbb{L}_i$, we get $\mathit{cost}(\mathbb{L}'_i) = \sum_{u\langle U\rangle \in \mathbb{L}'_i} \mathit{cost}(u\langle U\rangle) \leq \sum_{u\langle U\rangle \in \mathbb{L}'_i} \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_i)) = \mathit{cost}(\mathbb{L}_i)$. Using invariant 4 to bound the cost of the pebbling transitions $\mathbb{L}'_{i-1} \rightsquigarrow \mathbb{L}'_i$, we get the desired result $\mathit{cost}(\mathcal{L}') \leq 2 \cdot \mathit{cost}(\mathcal{L})$.

The construction is by forward induction over the moves in $\mathcal{L}$. Assume that the invariants hold for $\mathbb{L}_t$ and $\mathbb{L}'_t$.

Introduction $\mathbb{L}_{t+1} = \mathbb{L}_t \cup v\langle pred(v)\rangle$: If $v\langle pred(v)\rangle \preceq \mathbb{L}'_t$ we set $\mathbb{L}'_{t+1} = \mathbb{L}'_t$. For the subconfiguration $u\langle U\rangle \in \mathbb{L}'_t$ such that $v\langle pred(v)\rangle \preceq u\langle U\rangle$, we have $\mathit{cost}(u\langle U\rangle) \leq \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_t)) \leq \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_t \cup v\langle pred(v)\rangle))$, and for $u'\langle U'\rangle \in \mathbb{L}'_t$ distinct from $u\langle U\rangle$ nothing changes. All invariants stay true.

If $v\langle pred(v)\rangle \not\preceq \mathbb{L}'_t$, we introduce $v\langle pred(v)\rangle$ and expand to get $\mathbb{L}'_{t+1} = \mathsf{canon}(\mathbb{L}'_t \cup v\langle pred(v)\rangle)$. Invariants 1 and 2 obviously hold. We claim that invariant 3 holds with respect to $\mathbb{L}_{t+1}$ for all L-configurations $\mathbb{L}'$ in the transition $\mathbb{L}'_t \rightsquigarrow \mathbb{L}'_{t+1}$ upto and including $\mathbb{L}'_{t+1} = \mathsf{canon}(\mathbb{L}'_t \cup v\langle pred(v)\rangle)$. This claim yields invariants 3 and 4 for $\mathbb{L}'_{t+1}$.

To prove the claim, observe that invariant 3 holds for $\mathbb{L}'_t \cup v\langle pred(v)\rangle$ with respect to $\mathbb{L}_{t+1} = \mathbb{L}_t \cup v\langle pred(v)\rangle$ by the induction hypothesis and the fact that $\mathsf{proj}_{v\langle pred(v)\rangle}(\mathbb{L}_t \cup v\langle pred(v)\rangle) = v\langle pred(v)\rangle$. Since $\mathbb{L}'_{t+1}$ is obtained by repeated merging of non-overlapping subconfigurations from $\mathbb{L}'_t \cup v\langle pred(v)\rangle$, and since by induction over each such merger these subconfigurations meet the conditions in Lemma 7.18, the claim follows.

Expansion $\mathbb{L}_{t+3} = \big(\mathbb{L}_t \cup \mathsf{merge}(v_1\langle V_1\rangle, v_2\langle V_2\rangle)\big) \setminus \big\{v_1\langle V_1\rangle, v_2\langle V_2\rangle\big\}$: By induction $\mathbb{L}'_t \succeq \mathbb{L}_t \sim \mathbb{L}_{t+3}$, so there is a $u\langle U\rangle \in \mathbb{L}'_t$ such that $v_1\langle V_1\rangle, v_1\langle V_1\rangle \preceq u\langle U\rangle$. For $u'\langle U'\rangle \in \mathbb{L}'_t$ distinct from $u\langle U\rangle$ there are no changes, and if $\mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_{t+3})) \geq \mathit{cost}(u\langle U\rangle)$ nothing needs to be done and we can set $\mathbb{L}'_{t+3} = \mathbb{L}'_t$.

It can be the case, however, that the expansion within $\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_{t+3})$ decreased the cost so that $u\langle U\rangle$ is now too expensive. If so, we implode $u\langle U\rangle$ to a minimal non-wasteful L-configuration $\mathbb{M} \succeq \mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_{t+3})$ and set $\mathbb{L}'_{t+3} = \big(\mathbb{L}'_t \setminus u\langle U\rangle\big) \cup \mathbb{M}$.

Invariants 1 and 2 are immediate. Invariant 3 follows from Lemma 7.17 since $\mathbb{M}$ is chosen minimal. Thus, $\mathit{cost}(\mathbb{M}) \leq \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_{t+3}))$, and by the induction hypothesis we know that $\mathit{cost}(u\langle U\rangle) \leq \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_t))$. Using parts 1 and 2 of Proposition 7.5, we see that the implosion sequence $\mathbb{L}'_t \rightsquigarrow \mathbb{L}'_{t+3}$ causes an extra cost of at most

$$\begin{aligned}\mathit{cost}(u\langle U\rangle \cup \mathbb{M}) &\leq \mathit{cost}(u\langle U\rangle) + \mathit{cost}(\mathbb{M}) \\ &\leq \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_t)) + \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_{t+3})) \\ &\leq 2 \cdot \max_{i \in \{t, t+3\}} \big\{\mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_i))\big\},\end{aligned}$$

which yields invariant 4.

Implosion $\mathbb{L}_{t+m+1} = \big(\mathbb{L}_t \cup \mathbb{M}\big) \setminus v\langle V\rangle$ for $\mathbb{M} = \big\{v_i\langle V_i\rangle \mid i \in [m]\big\}$: This case is completely analogous to the expansion case. Again $v\langle V\rangle$ is covered by some $u\langle U\rangle \in \mathbb{L}'_t$, and if $\mathit{cost}(u\langle U\rangle) > \mathit{cost}(\mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_{t+m+1}))$ we implode $u\langle U\rangle$ to a minimal non-wasteful $\mathbb{M} \succeq \mathsf{proj}_{u\langle U\rangle}(\mathbb{L}_{t+m+1})$ and set $\mathbb{L}'_{t+m+1} = \big(\mathbb{L}'_t \setminus u\langle U\rangle\big) \cup \mathbb{M}$. Using Lemma 7.17 and Proposition 7.5, we get invariants 1-4.

Going through the moves in $\mathcal{L} = \big\{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\big\}$, this construction yields an L-pebbling $\mathcal{L}' = \big\{\mathbb{L}'_0, \ldots, \mathbb{L}'_{\tau'}\big\}$ without wasteful implosions such that $\mathbb{L}'_{\tau'} \succeq \mathbb{L}_\tau$ and $\mathit{cost}(\mathcal{L}') \leq 2 \cdot \mathit{cost}(\mathcal{L})$. $\square$

Thereby, the proof of Lemma 6.8 outlined in the beginning of this section is complete. We repeat the proof in condensed form for completeness.

Proof of Lemma 6.8. Let $\mathcal{L}$ be an arbitrary L-pebbling of T. By Observation 7.1, we can assume $\mathcal{L}$ to be non-redundant. Using Lemma 7.9, we get a non-overlapping pebbling $\mathcal{L}'$ with $\mathsf{cost}(\mathcal{L}') \leq \mathsf{cost}(\mathcal{L})$. If $\mathcal{L}'$ contains wasteful implosions, Lemma 7.19 yields a non-wasteful pebbling $\mathcal{L}''$ in $\mathsf{cost}(\mathcal{L}'') \leq 2 \cdot \mathsf{cost}(\mathcal{L}')$. Finally, Corollary 7.14 transforms $\mathcal{L}''$ into a reversal-free L-pebbling $\mathcal{L}'''$ of T such that $\mathsf{cost}(\mathcal{L}''') \leq \mathsf{cost}(\mathcal{L}'') \leq 2 \cdot \mathsf{cost}(\mathcal{L})$. The lemma follows. $\square$

8 Resolution Derivations Induce Labelled Pebblings

In this section, we shift our focus to resolution and show that clause configurations can be interpreted in terms of labelled pebble configurations in such a way that resolution derivations induce legal L-pebblings. We first give some technical preliminaries. Then we try to explain the intuition for how sets of clauses are translated into sets of pebbles. Finally, we state the formal definitions and prove the correspondence between resolution derivations and L-pebblings.

We start with the technicalities. For simplicity, in the following we will write $v_1, \ldots, v_d$ instead of $x(v)_1, \ldots, x(v)_d$ for the d variables associated with the vertex v in a dth degree pebbling contradiction.

Definition 8.1. Assume that G is a DAG with a unique target z and all vertices having indegree 0 or 2. Then we define $^*Peb^d_G = Peb^d_G \setminus \{\overline{z}_1, \ldots, \overline{z}_d\}$ to be the pebbling contradiction with target axioms removed.

Our first observation is that instead of refutations of Peb^d_G, we may study derivations of $\bigvee_{i=1}^{d} z_i$ from $^*Peb^d_G$.

Observation 8.2. *For any DAG G with a unique target z and all vertices having indegree 0 or 2, it holds that $Sp\big(Peb^d_G \vdash 0\big) = Sp\big({}^*Peb^d_G \vdash \bigvee_{l=1}^{d} z_l\big)$.*

Proof. For any resolution derivation $\pi^* : {}^*Peb^d_G \to \bigvee_{l=1}^{d} z_l$, we can get a resolution refutation of Peb^d_G from π^* in the same space by resolving $\bigvee_{l=1}^{d} z_l$ with all $\overline{z}_l$, $l = 1, \ldots, d$, in space 3. In the other direction, for $\pi : Peb^d_G \to 0$ we can extract a derivation of $\bigvee_{l=1}^{d} z_l$ in at most the same space by simply omitting all downloads of and resolution steps on $\overline{z}_l$ in π, leaving the literals z_l in the clauses. Instead of the final empty clause 0 we get some clause $D \subseteq \bigvee_{l=1}^{d} z_l$, and since ${}^*Peb^d_T \nvDash D \subsetneqq \bigvee_{l=1}^{d} z_l$ and resolution is sound, we have $D = \bigvee_{l=1}^{d} z_l$. $\square$

The following easy lemma will be used repeatedly.

Lemma 8.3. *Suppose that C, D are clauses and $\mathbb{C}, \mathbb{D}$ sets of clauses.*

1. $\mathbb{C} \cup \{C\} \vDash D$ *if and only if* $\mathbb{C} \vDash \overline{a} \vee D$ *for all* $a \in Lit(C)$.

2. $\mathbb{C} \cup \mathbb{D} \vDash D$ *for* $\mathbb{D} = \{D_1, \ldots, D_m\}$ *if and only if* $\mathbb{C} \vDash \bigvee_{i \in [m]} \overline{a}_i \vee D$ *for all choices of literals* $(a_1, \ldots, a_m) \in Lit(D_1) \times \cdots \times Lit(D_m)$.

Proof. For part 1, assume that $\mathbb{C} \cup \{C\} \vDash D$ and consider an assignment α such that $\alpha(\mathbb{C}) = 1$ and $\alpha(D) = 0$ (if there is no such α, then $\mathbb{C} \vDash D \subseteq \overline{a} \vee D$). Such an α sets all $\overline{a}$ to true. Conversely, if $\mathbb{C} \vDash \overline{a} \vee D$ for all $a \in \mathit{Lit}(C)$ and α is such that $\alpha(\mathbb{C}) = \alpha(C) = 1$, it must hold that $\alpha(D) = 1$.

Part 2 follows from part 1 by induction. ☐

We introduce some space-saving notation. If $\mathit{pred}(r) = \{p, q\}$ we say that the *axioms for* r in $^*\mathit{Peb}^d_G$ is the set $Ax^d(r) = \{\overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^d r_l \mid i, j \in [d]\}$. If r is a source, we define $Ax^d(r) = \{\bigvee_{i=1}^d r_i\}$. For V a set of vertices, let $Ax^d(V) = \{Ax^d(v) \mid v \in V\}$.

For v a vertex in T, we let $\mathbb{B}(v) = \bigvee_{i=1}^d v_i$. For $V \subseteq V(T)$, we define $\mathbb{B}(V) = \{\mathbb{B}(v) \mid v \in V\}$ and $A_V = \bigvee_{v \in V} \bigvee_{i=1}^d v_i$. $\mathbb{B}(V)$ can be understood as "truth of all vertices in V" and A_V as "truth of some vertex in V".

This concludes the technical preliminaries. We next try to provide some intuition for how clause configurations are translated into pebble configurations.

Let us associate each vertex $v \in V(T)$ with the clauses $Ax^d(v)$. In the standard black-white pebble game, if at some time t there is an independent black pebble on v, an optimal pebbling will not pebble any vertex in T^v after time t. As an analogy of this, a clause configuration $\mathbb{C}_t$ should induce an independent black pebble on v only if no axioms from $Ax^d(T^v) = {^*\mathit{Peb}^d_{T^v}}$ need be used to derive $\bigvee_{l=1}^d z_l$. This holds if and only if

$$\mathbb{C} \cup \left({^*\mathit{Peb}^d_T} \setminus {^*\mathit{Peb}^d_{T^v}}\right) \vDash \bigvee_{l=1}^d z_l \tag{2}$$

by the implicational completeness of resolution. If (2) holds for v but not for $\mathit{succ}(v)$, we can interpret this by saying that the resolution derivation "has reached as far as v but not any farther" and indicate this fact by placing an independent black pebble on v.

It turns out that (2) is equivalent with the condition that $\mathbb{C}$ together with the truth of all vertices unrelated to v should imply truth of some vertex on the path from v to the root, or more concisely

$$\mathbb{C} \cup \mathbb{B}\left(T \setminus \left(T^v \cup P^v\right)\right) \vDash A_{P^v}\,, \tag{3}$$

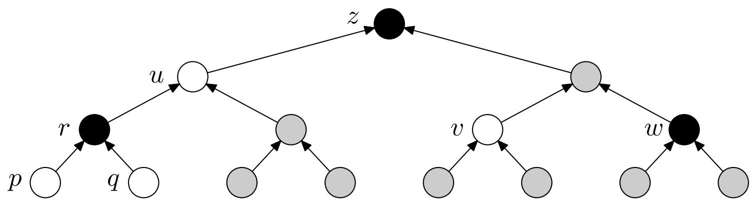
and the condition (3) is more convenient to work with. In the next lemma, we prove the equivalence of (2) and (3). The lemma is intended only as a way to strengthen the intuition and motivate the formal definitions below. It will not be used in the following and is therefore optional reading.

Lemma 8.4. *Suppose that the clause configuration $\mathbb{C}$ is derived from $^*\mathit{Peb}^d_T$ for a complete binary tree T with root z and let r be an arbitrary vertex in $V(T)$. Then $\mathbb{C} \cup \mathbb{B}\left(T \setminus \left(T^r \cup P^r\right)\right) \vDash A_{P^r}$ if and only if $\mathbb{C} \cup \left({^*\mathit{Peb}^d_T} \setminus {^*\mathit{Peb}^d_{T^r}}\right) \vDash \bigvee_{l=1}^d z_l$.*

Proof. Note first that if $r = z$, the two implications are exactly the same. Assume therefore that r is not the root and that it has sibling s and successor u.

($\Rightarrow$) Suppose that $\mathbb{C} \cup \mathbb{B}\left(T \setminus \left(T^r \cup P^r\right)\right) \vDash A_{P^r}$. For all $v \in T \setminus \left(T^r \cup P^r\right)$ it holds that ${^*\mathit{Peb}^d_T} \setminus {^*\mathit{Peb}^d_{T^r}} \vDash \bigvee_{l=1}^d v_l$, since ${^*\mathit{Peb}^d_{T^v}} \subseteq {^*\mathit{Peb}^d_T} \setminus {^*\mathit{Peb}^d_{T^r}}$ and ${^*\mathit{Peb}^d_{T^v}} \vDash \bigvee_{l=1}^d v_l$, and the fact that resolution is implicationally complete means that these clauses are all derivable. Write $A_{P^r} = A_{P^r_*} \vee \bigvee_{i=1}^d r_i$. Resolve with $\overline{r}_i \vee \overline{s}_j \vee \bigvee_{l=1}^d u_l$ for all $i, j \in [d]$ to get $\{\overline{s}_j \vee A_{P^r_*} \mid j \in [d]\}$, derive $\bigvee_{j=1}^d s_j$ by

$$\mathbb{C} = \left\{\overline{u}_i \vee \overline{v}_j \vee \bigvee_{l=1}^{d} z_l,\ \overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l,\ \bigvee_{l=1}^{d} w_l \mid 1 \le i,j \le d\right\}$$



$$\mathbb{L}(\mathbb{C}) = \{z\langle u, v\rangle,\ r\langle p, q\rangle,\ w\langle\emptyset\rangle\}$$

Figure 9: An example clause configuration $\mathbb{C}$ and induced L-configuration $\mathbb{L}(\mathbb{C})$.

implicational completeness and then resolve the clauses $\left\{\overline{s}_j \vee A_{P_*^r} \mid j \in [d]\right\}$ and $\bigvee_{j=1}^{d} s_j$ to get $A_{P_*^r}$. In the same way we can eliminate all vertices in $P^r \setminus \{z\}$ from $A_{P_*^r}$ and derive $\bigvee_{l=1}^{d} z_l$ using only axioms from $*Peb_T^d \setminus *Peb_{T^r}^d$. Since resolution is sound this implies that $\mathbb{C} \cup \left(*Peb_T^d \setminus *Peb_{T^r}^d\right) \vDash \bigvee_{l=1}^{d} z_l$.

($\Leftarrow$) Rewrite the assumption as

$$\mathbb{C} \cup Ax^d\big(T \setminus (T^r \cup P^r)\big) \cup Ax^d\big(P_*^u\big) \cup \left\{\overline{r}_i \vee \overline{s}_j \vee \bigvee_{l=1}^{d} u_l \mid i,j \in [d]\right\} \vDash \bigvee_{l=1}^{d} z_l.$$

Repeated use of Lemma 8.3 yields

$$\mathbb{C} \cup Ax^d\big(T \setminus (T^r \cup P^r)\big) \cup Ax^d(P_*^u) \vDash \bigvee_{l=1}^{d} r_l \vee \bigvee_{l=1}^{d} z_l$$

and proceeding in the same way for all $w \in P_*^u$ we get

$$\mathbb{C} \cup Ax^d\big(T \setminus (T^r \cup P^r)\big) \vDash \bigvee_{v \in P^r} \bigvee_{l=1}^{d} v_l = A_{P^r}.$$

Any α satisfying $\mathbb{B}\big(T \setminus (T^r \cup P^r)\big)$ must satisfy $Ax^d\big(T \setminus (T^r \cup P^r)\big)$ and thus

$$\mathbb{C} \cup \mathbb{B}\big(T \setminus (T^r \cup P^r)\big) \vDash A_{P^r}. \qquad \square$$

Continuing our intuitive argument, the simplest case for a black pebble on a vertex v is when $\mathbb{C} \vDash \bigvee_{i=1}^{d} v_i$. Let us restrict our attention to this case and think of a black pebble on v as derived truth $\mathbb{B}(v) = \bigvee_{i=1}^{d} v_i$ of v. One way of looking at a dependent black pebble on v supported by white pebbles on W, or, in L-pebbling terminology, a subconfiguration $v\langle W\rangle$, is that given independent black pebbles on all $w \in W$ we can eliminate the white pebbles and get $v\langle\emptyset\rangle$. By analogy, a clause configuration $\mathbb{C}$ should induce a subconfiguration $v\langle W\rangle$ if we would get an induced independent black pebble on v by assuming the truth of all $w \in W$, i.e., if $\mathbb{C} \cup \mathbb{B}(W) \vDash \bigvee_{i=1}^{d} v_i$. Figure 9 (which is Figure 3 but with renamed vertices) gives an example of this intuitive understanding of induced pebble configurations.

Our formal definitions follow the intuition presented above quite closely, modulo a few technical details.

Definition 8.5 (Support). Suppose for $\mathbb{C}$ a set of clauses, $v \in V(T)$ a vertex and $V \subseteq T \setminus P^v$ a set of vertices that $\mathbb{C} \cup \mathbb{B}(V) \vDash A_{P^v}$. Then V is a *support* for v with respect to $\mathbb{C}$, and if there is no $V' \subsetneqq V$ such that $\mathbb{C} \cup \mathbb{B}\bigl(V'\bigr) \vDash A_{P^v}$ the support is *minimal*. If V is a minimal support for v with respect to $\mathbb{C}$ such that $\mathbb{C} \cup \mathbb{B}\bigl(V\bigr) \nvDash A_{P^v_*}$, we say that v is *maximal* with respect to $\mathbb{C}$ and V.

For V a support of v, we define the *supporting white pebbles* of v to be $\mathit{swp}(v, V) = \bigl\{w \in V \cap T^v_* \mid P^w_* \cap V = \emptyset\bigr\}$.

When it is clear from context, we sometimes omit which support or vertex is minimal or maximal with respect to what. Note that $\mathit{swp}(v, V)$ is a simple roof below v over $V \cap T^v_*$.

Definition 8.6 (Induced L-configuration). For $\mathbb{C}$ a set of clauses derived from $^*\mathit{Peb}^d_T$, the *induced L-configuration* $\mathbb{L}(\mathbb{C})$ consists of all subconfigurations $v\langle V\rangle$ such that

1. there is a minimal support $V' \subseteq T \setminus P^v$ for v with respect to $\mathbb{C}$,
2. v is maximal with respect to $\mathbb{C}$ and V',
3. $V = \mathit{swp}(v, V')$.

Remark 8.7. The reason we use $V = \mathit{swp}(v, V')$ instead of $V' \cap T^v_*$ is that we need simple sets (Definition 6.3) to define our induced subconfigurations $v\langle V\rangle$, but the supporting sets V' are not necessarily simple. For instance, if we let

$$\mathbb{C}' = \bigl\{\overline{u}_i \vee \overline{v}_j \vee \overline{q}_l \vee \textstyle\bigvee_{n=1}^d z_n,\ \overline{p}_i \vee \overline{q}_j \vee \textstyle\bigvee_{n=1}^d r_n,\ \textstyle\bigvee_{n=1}^d w_n \mid 1 \leq i, j, l \leq d\bigr\}$$

in Figure 9, the root z has the minimal supporting set $V' = \{u, v, q\}$. For technical reasons, it is simpler to ignore all but the topmost vertices in V', so by Definition 8.6 we get $\mathbb{L}(\mathbb{C}') = \mathbb{L}(\mathbb{C})$. Anyway, it seems very plausible that optimal resolution derivations should never result in clause configurations like $\mathbb{C}'$, and since the bound we will prove is asymptotically tight we see that we do not lose anything by restricting the white pebbles to $V = \mathit{swp}(v, V')$ instead of $V' \cap T^v_*$.

Note also that a black pebble on v is defined in terms of A_{P^v}, not $\bigvee_{i=1}^d v_i$. This means that for instance

$$\mathbb{C}'' = \bigl\{\overline{u}_i \vee \overline{v}_j \vee \textstyle\bigvee_{n=1}^d z_n, \overline{p}_i \vee \overline{q}_j \vee \textstyle\bigvee_{n=1}^d r_n, \textstyle\bigvee_{n=1}^d w_n \vee \textstyle\bigvee_{n=1}^d z_n \mid 1 \leq i, j \leq d\bigr\}$$

also induces an independent black pebble $w\langle\emptyset\rangle$, and $\mathbb{L}(\mathbb{C}'') = \mathbb{L}(\mathbb{C})$.

Recall that the goal of this section is to show that resolution derivations induce L-pebblings. Suppose that $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is a resolution derivation of $\bigvee_{l=1}^d z_l$ from $^*\mathit{Peb}^d_T$. For $\mathbb{C}_0 = \emptyset$ we obviously get $\mathbb{L}(\mathbb{C}_0) = \emptyset$, and it is not hard to see that at the end of the derivation $\mathbb{C}_\tau = \bigl\{\bigvee_{n=1}^d z_n\bigr\}$ induces a single independent black pebble $\mathbb{L}(\mathbb{C}_\tau) = \bigl\{z\langle\emptyset\rangle\bigr\}$ on the root of T. Hence, we are done if we can prove that $\bigl\{\mathbb{L}(\mathbb{C}_0), \ldots \mathbb{L}(\mathbb{C}_\tau)\bigr\}$ forms the backbone of a legal L-pebbling $\mathcal{L}$, where the transitions $\mathbb{L}(\mathbb{C}_t) \rightsquigarrow \mathbb{L}(\mathbb{C}_{t+1})$ can be accomplished in accordance with the rules of the L-pebble game.

By the L-pebbling rules in Definition 6.7, any subconfiguration $v\langle V\rangle$ may be erased from $\mathcal{L}$ freely at any time. Consequently, we need not worry about subconfigurations $v\langle V\rangle \in \mathbb{L}(\mathbb{C}_t) \setminus \mathbb{L}(\mathbb{C}_{t+1})$ disappearing during the transition from $\mathbb{C}_t$ to $\mathbb{C}_{t+1}$. What we do need to check, though, is that no $v\langle V\rangle$ suddenly appears inexplicably in $\mathbb{L}(\mathbb{C}_{t+1})$ as a result of a resolution derivation step $\mathbb{C}_i \rightsquigarrow \mathbb{C}_{i+1}$, but that we can always derive any $v\langle V\rangle \in \mathbb{L}(\mathbb{C}_{t+1}) \setminus \mathbb{L}(\mathbb{C}_t)$ from $\mathbb{L}(\mathbb{C}_t)$ by the L-pebbling rules.

The rest of this section is devoted to proving this. We first make a pair of observations. The first observation relates subset containment of supporting sets and the order relation between corresponding subconfigurations.

Observation 8.8. *If v, $u \in P^v$ and $U', V' \subseteq T \setminus P^v$ are vertices and sets such that $U' \cap T^v_* \subseteq V' \cap T^v_*$, then $u\langle swp(u, U')\rangle \succeq v\langle swp(v, V')\rangle$.*

Proof. Using the characterization of $\preceq$ in Observation 6.6 on page 17, it is sufficient to prove that $v \in T^u$ and $P^v \cap swp(u, U') = \emptyset$ and that $swp(v, V')$ is a simple roof below v over $swp(u, U') \cap T^v$.

The condition $v \in T^u$ is equivalent to $u \in P^v$, and since $U' \subseteq T \setminus P^v$ it clearly holds that $P^v \cap swp(u, U') \subseteq P^v \cap U' = \emptyset$. Both $swp(v, V')$ and $swp(u, U') \cap T^v$ are simple sets below v by assumption. The only nontrivial part is to establish that $swp(v, V')$ is a roof over $swp(u, U') \cap T^v$.

Suppose that $w \in swp(u, U') \cap T^v$. To prove that $swp(v, V')$ is a roof, we need to find a $w' \in P^w \cap swp(v, V')$. Since by assumption $swp(u, U') \cap T^v \subseteq U' \cap T^v_* \subseteq V' \cap T^v_*$, it holds that $w \in V' \cap T^v_*$. If $w \in swp(v, V')$ we are done, so suppose $w \notin swp(v, V')$. The reason that w is missing from $swp(v, V')$ must be that $P^w_* \cap V' \neq \emptyset$, but if we pick $w' \in P^w_* \cap V'$ of maximal height we get $P^{w'}_* \cap V' = \emptyset$ and $w' \in swp(v, V')$. This w' satisfies $w \in P^w \cap swp(v, V')$, which shows that $swp(v, V')$ is a roof over $swp(u, U') \cap T^v$. ☐

The second observation says that if a support V' is not minimal or a vertex v is not maximal with respect to a clause configuration $\mathbb{C}$, then this just means that $\mathbb{C}$ induces something stronger than $v\langle swp(v, V')\rangle$.

Observation 8.9. *If $\mathbb{C} \cup \mathbb{B}(V') \vDash A_{P^v}$ for $V' \subseteq T \setminus P^v$, then there is a subconfiguration $u\langle U\rangle \in \mathbb{L}(\mathbb{C})$ such that $v\langle swp(v, V')\rangle \preceq u\langle U\rangle$.*

Proof. Minimize $U' \subseteq V'$ and then maximize $u \in P^v$ so that $\mathbb{C} \cup \mathbb{B}(U') \vDash A_{P^u}$. Set $U = swp(u, U')$ and use Observation 8.8. ☐

With the help of these observations we can analyze how new subconfigurations $v\langle V\rangle$ may appear in $\mathbb{L}(\mathbb{C}_{t+1})$ after a resolution derivation step $\mathbb{C}_i \rightsquigarrow \mathbb{C}_{i+1}$.

Observation 8.10 (Inference). *If $\mathbb{C}_{t+1}$ is derived from $\mathbb{C}_t$ by inference, then $\mathbb{L}(\mathbb{C}_{t+1}) = \mathbb{L}(\mathbb{C}_t)$.*

Proof. $\mathbb{C}_t$ and $\mathbb{C}_{t+1}$ have the same logical consequences. ☐

Lemma 8.11 (Erasure). *Suppose that $\mathbb{C}_{t+1}$ is derived from $\mathbb{C}_t$ by erasure. Then for each $v\langle V\rangle \in \mathbb{L}(\mathbb{C}_{t+1})$ there is a $u\langle U\rangle \in \mathbb{L}(\mathbb{C}_t)$ such that $v\langle V\rangle \preceq u\langle U\rangle$.*

Proof. By assumption there is a $V' \subseteq T \setminus P^v$ such that $V = swp(v, V')$ and $\mathbb{C}_{t+1} \cup \mathbb{B}(V') \vDash A_{P^v}$. Certainly, the same implication holds for $\mathbb{C}_t \supseteq \mathbb{C}_{t+1}$. The lemma follows from Observation 8.9. ☐

In particular, all new subconfigurations resulting from an erasure $\mathbb{C}_t \rightsquigarrow \mathbb{C}_{t+1}$ can be obtained from $\mathbb{L}(\mathbb{C}_t)$ by reversal. One way of interpreting this is that no white pebbles can just disappear at an erasure step except if the black pebble that they support disappear as well. This is exactly the kind of "controlled removal" of white pebbles that the L-pebble game was designed to capture.

Lemma 8.12 (Axiom download). *If $\mathbb{C}_{t+1} = \mathbb{C}_t \cup \{C\}$ for an axiom clause $C \in Ax^d(r)$, then all subconfigurations $v\langle V\rangle \in \mathbb{L}(\mathbb{C}_{t+1}) \setminus \mathbb{L}(\mathbb{C}_t)$ can be obtained from $\mathbb{L}(\mathbb{C}_t) \cup r\langle pred(r)\rangle$ by reversals from subconfigurations in $\mathbb{L}(\mathbb{C}_t)$ followed by mergers on $\{r\} \cup pred(r)$.*

Proof. By assumption, there is a minimal $V' \subseteq T \setminus P^v$ with $V = swp(v, V')$ such that $\mathbb{C}_t \cup \{C\} \cup \mathbb{B}(V') \vDash A_{P^v}$ for $C \in Ax^d(r)$. We will use repeatedly the fact that $\mathbb{B}(r) \vDash C$.

It is intuitively clear that axioms $C \in Ax^d(r)$ should not yield any interesting new subconfigurations $v\langle V\rangle$ if $r \in T \setminus T^v$, and for $r \in T^v$ we should be able to explain new subconfigurations with the help of $r\langle pred(r)\rangle$. We prove this by a case analysis over r.

$r \in T \setminus (T^v \cup P^v)$: We have $\mathbb{C}_t \cup \mathbb{B}(V' \cup \{r\}) \vDash A_{P^v}$ for $V' \cup \{r\} \subseteq T \setminus P^v$, so Observation 8.9 tells us that there is a $u\langle U\rangle \in \mathbb{L}(\mathbb{C}_t)$ such that $v\langle V\rangle = v\langle swp(v, V')\rangle = v\langle swp(v, V' \cup \{r\})\rangle \preceq u\langle U\rangle$.

$r \in P^v_*$: Write $C = \overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l$ for $\{p, q\} = pred(r) \neq \emptyset$ and let p be the vertex in $P^v \cap pred(r)$. Using Lemma 8.3 to move p_i to the right of the implication sign yields $\mathbb{C}_t \cup \mathbb{B}(V') \vDash A_{P^v} \vee p_i = A_{P^v}$, and since V' is minimal it follows that $v\langle V\rangle \in \mathbb{L}(\mathbb{C}_t)$.

$r = v$: Note first that we are prepared to accept the introduction of $r\langle pred(r)\rangle$ without any explanation, so if $\mathbb{C}_t \cup \{C\} \cup \mathbb{B}(V') \vDash A_{P^r}$ for $pred(r) \subseteq V'$ no further analysis is needed for $r\langle swp(r, V')\rangle = r\langle pred(r)\rangle$. In particular, this is always the case if $pred(r) = \emptyset$, i.e., if r is a source.

Suppose that $v\langle V\rangle = r\langle swp(r, V')\rangle \in \mathbb{L}(\mathbb{C}_{t+1})$ for $V \neq pred(r) = \{p, q\}$, and write $C = \overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l$. We want to derive $r\langle V\rangle$ by the pebbling rules from $\mathbb{L}(\mathbb{C}_r) \cup r\langle pred(r)\rangle$. By symmetry, we get two subcases.

1. $p \in V, q \notin V$: By Definition 8.5, we have $p \in V'$ and $q \notin V'$. Observe that this implies that $V' \subseteq T \setminus P^q$. Also, we can use Lemma 8.3 to move q_j to the right-hand side of the implication sign and get $\mathbb{C}_t \cup \mathbb{B}(V') \vDash A_{P^r} \vee q_j \subseteq A_{P^r} \vee \bigvee_{j=1}^{d} q_j = A_{P^q}$. Plugging this into Observation 8.9 shows that there is a $w\langle W\rangle \in \mathbb{L}(\mathbb{C}_t)$ such that $q\big\langle V \setminus \{p\}\big\rangle = q\big\langle swp(q, V')\big\rangle \preceq w\langle W\rangle$. Thus we can derive $q\big\langle V \setminus \{p\}\big\rangle$ from $\mathbb{L}(\mathbb{C}_t)$ by reversal and then merge $r\langle pred(r)\rangle = r\langle p, q\rangle$ with $q\big\langle V \setminus \{p\}\big\rangle$ to obtain $r\big\langle (\{p, q\} \cup (V \setminus \{p\})) \setminus \{q\}\big\rangle = r\langle V\rangle$.
2. $p, q \notin V$: Again by Definition 8.5, we have $p, q \notin V'$. If we use Lemma 8.3 twice we get $\mathbb{C}_t \cup \mathbb{B}(V') \vDash A_{P^p} \wedge A_{P^q}$, and noting that $V' \subseteq T \setminus (P^p \cup P^p)$ we can apply Observation 8.9 to derive $p\big\langle V \cap T^p_*\big\rangle$ and $q\big\langle V \cap T^q_*\big\rangle$ from $\mathbb{L}(\mathbb{C}_t)$ by reversals. Merging these subconfigurations with $r\langle p, q\rangle$, we get $r\big\langle (V \cap T^p_*) \cup (V \cap T^q_*)\big\rangle = r\langle V\rangle$.

$r \in T^v_*$: By assumption, $\mathbb{C}_t \cup \{C\} \cup \mathbb{B}(V') \vDash A_{P^v}$, and since $r \in T^v_*$ and $\mathbb{B}(r) \vDash C$ we have $\mathbb{C}_t \cup \mathbb{B}(V' \cup \{r\}) \vDash A_{P^v}$ for $V' \cup \{r\} \subseteq T \setminus P^v$. If $P^r \cap V' \neq \emptyset$, it holds that $swp(v, V' \cup \{r\}) = swp(v, V')$ and we can obtain $v\langle V\rangle$ from $\mathbb{L}(\mathbb{C}_t)$ by reversal according to Observation 8.9, so suppose $P^r \cap V' = \emptyset$.

Pick $U' \subseteq V' \cup \{r\}$ minimal and then $u \in P^v$ maximal with respect to U' such that $\mathbb{C}_t \cup \mathbb{B}(U') \vDash A_{P^u}$. By the minimality of V' we have

$r \in U'$, and since $P^r_* \cap U' \subseteq P^r_* \cap V' = \emptyset$ it holds that $r \in \mathit{swp}(u, U')$. Consequently, we cannot use $u\langle U\rangle = u\big\langle \mathit{swp}(u, U')\big\rangle \in \mathbb{L}(\mathbb{C}_t)$ to derive $v\langle V\rangle \not\preceq u\langle U\rangle$ by reversal. However, since $U' \subseteq V' \cup \{r\}$, Observation 8.8 tells us that $v\big\langle (V \cup \{r\}) \setminus T^r_*\big\rangle = v\big\langle \mathit{swp}(v, V' \cup \{r\})\big\rangle \preceq u\langle U\rangle$ can be derived by reversal from $\mathbb{L}(\mathbb{C}_t)$. If we could also derive $r\big\langle V \cap T^r_*\big\rangle$ from $\mathbb{L}(\mathbb{C}_t) \cup r\langle \mathit{pred}(r)\rangle$, a merger would produce the desired subconfiguration $v\Big\langle \Big(\big((V \cup \{r\}) \setminus T^r_*\big) \cup \big(V \cap T^r_*\big)\Big) \setminus \{r\}\Big\rangle = v\langle V\rangle$.

Hence, we are done if we can derive $r\big\langle V \cap T^r_*\big\rangle = r\big\langle \mathit{swp}(v, V') \cap T^r_*\big\rangle = r\big\langle \mathit{swp}(r, V')\big\rangle$ from $\mathbb{L}(\mathbb{C}_t) \cup r\langle \mathit{pred}(r)\rangle$. But $A_{P^r} \supseteq A_{P^v}$, so by assumption we have $\mathbb{C}_t \cup \{C\} \cup \mathbb{B}\big(V'\big) \vDash A_{P^r}$ for $V' \subseteq T \setminus P^r$. This is almost exactly the case $r = v$ above, where we proved that $r\big\langle \mathit{swp}(r, V')\big\rangle$ is derivable from $\mathbb{L}(\mathbb{C}_t) \cup r\langle \mathit{pred}(r)\rangle$. The only difference is that now it is not necessarily true that V' is a minimal support and that r is maximal with respect to V'. But these assumptions were not used in the derivation of $r\big\langle \mathit{swp}(r, V')\big\rangle$ from $\mathbb{L}(\mathbb{C}_t) \cup r\langle \mathit{pred}(r)\rangle$ anyway, so we can reuse exactly the same proof here to get $r\big\langle \mathit{swp}(r, V')\big\rangle$. This concludes the analysis for $r \in T^v_*$.

Studying the pebbling moves in the case analysis above, we see that all subconfigurations $v\langle V\rangle \in \mathbb{L}(\mathbb{C}_{t+1}) \setminus \mathbb{L}(\mathbb{C}_t)$ can be obtained from $\mathbb{L}(\mathbb{C}_t) \cup r\langle \mathit{pred}(r)\rangle$ by a (possibly empty) sequence of reversals from $\mathbb{L}(\mathbb{C}_t)$, followed by a (possibly empty) sequence of mergers on $\{r\} \cup \mathit{pred}(r)$. ☐

Combining the results proven for axiom download, inference and erasure, we can show that a resolution derivation induces a legal L-pebbling.

Theorem 8.13. *Let $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ be a resolution derivation of $\bigvee_{l=1}^{d} z_l$ from *Peb^d_T. Then $\big\{\mathbb{L}(\mathbb{C}_0), \ldots, \mathbb{L}(\mathbb{C}_\tau)\big\}$ is the backbone of a legal L-pebbling $\mathcal{L}$ of T such that $\max_{t \in [\tau]}\big\{\mathit{cost}(\mathbb{L}(\mathbb{C}_t))\big\} = \Omega\big(\mathit{cost}(\mathcal{L})\big)$.*

Proof. The fact that $\big\{\mathbb{L}(\mathbb{C}_0), \ldots, \mathbb{L}(\mathbb{C}_\tau)\big\}$ essentially is a legal L-pebbling was proven in Observation 8.10, Lemma 8.11 and Lemma 8.12, where it was explicitly indicated how the "holes" in $\mathbb{L}(\mathbb{C}_t) \rightsquigarrow \mathbb{L}(\mathbb{C}_{t+1})$ could be filled in by L-pebbling moves to get a legal pebbling $\mathcal{L}$.

The bound $\max_{t \in [\tau]}\big\{\mathit{cost}(\mathbb{L}(\mathbb{C}_t))\big\} = \Omega\big(\mathit{cost}(\mathcal{L})\big)$ does not follow immediately from this, however. The problem is that a single resolution derivation step $\mathbb{C}_t \rightsquigarrow \mathbb{C}_{t+1}$ may induce several L-pebbling moves to get from $\mathbb{L}(\mathbb{C}_t)$ to $\mathbb{L}(\mathbb{C}_{t+1})$ in $\mathcal{L}$. Therefore, we have to consider the possibility[3] that the maximal pebbling cost in $\mathcal{L}$ is reached in some intermediate L-configuration $\mathbb{L}'$ in between $\mathbb{L}(\mathbb{C}_t)$ and $\mathbb{L}(\mathbb{C}_{t+1})$.

Since inference steps in π do not change the set of induced L-configurations, we get two cases.

1. $\mathbb{C}_t \rightsquigarrow \mathbb{C}_{t+1}$ is an erasure. The moves to get from $\mathbb{L}(\mathbb{C}_t)$ to $\mathbb{L}(\mathbb{C}_{t+1})$ are a series of reversals from $\mathbb{L}(\mathbb{C}_t)$ followed by a series of erasures from $\mathbb{L}(\mathbb{C}_t)$. In view of part 1 of Proposition 7.5, without loss of generality we can let $\mathbb{L}'$ be the L-configuration after all reversals but before all erasures. Then $\mathbb{L}' = \mathbb{L}(\mathbb{C}_t) \cup \mathbb{L}(\mathbb{C}_{t+1})$, and by part 2 of Proposition 7.5, we have $\mathit{cost}(\mathbb{L}') \leq \mathit{cost}(\mathbb{L}(\mathbb{C}_t)) + \mathit{cost}(\mathbb{L}(\mathbb{C}_{t+1})) \leq 2 \cdot \max_{i \in [t, t+1]}\big\{\mathit{cost}(\mathbb{L}(\mathbb{C}_i))\big\}$.

[3] In fact, this does not happen, but instead of proving this we happily sacrifice a constant 2 here in order to get a simpler (or at least slightly less involved) proof.

2. $\mathbb{C}_t \rightsquigarrow \mathbb{C}_{t+1}$ is a download of $C \in Ax^d(v)$. In this case the sequence of moves to get from $\mathbb{L}(\mathbb{C}_t)$ to $\mathbb{L}(\mathbb{C}_{t+1})$ is a possible introduction of $v\langle pred(v)\rangle$ followed by a series of reversals from $\mathbb{L}(\mathbb{C}_t)$, then a series of mergers on $\{v\} \cup pred(v)$ and finally a series of erasures of subconfigurations not derived in the merger moves. Again by part 1 of Proposition 7.5, we may let $\mathbb{L}'$ be the L-configuration after all mergers but before the erasures.

   All pebbles in $Bl(\mathbb{L}') \cup Wh(\mathbb{L}')$ are present in either $\mathbb{L}(\mathbb{C}_t)$ or $\mathbb{L}(\mathbb{C}_{t+1})$, except possibly for the pebbles on $\{v\} \cup pred(v)$ which may have been introduced and then merged away. Since by construction all subconfigurations resulting from these mergers must be contained in $\mathbb{L}(\mathbb{C}_{t+1})$, the pebbles on $\{v\} \cup pred(v)$ are the only ones that can appear and then disappear during the intermediate pebbling steps. If we remove $\{v\} \cup pred(v)$ from $Bl(\mathbb{L}') \cup Wh(\mathbb{L}')$ the pebbling cost cannot decrease by more than 3.

   Since all pebbles $Bl(\mathbb{L}') \setminus (\{v\} \cup pred(v))$ and $Wh(\mathbb{L}') \setminus (\{v\} \cup pred(v))$ are contained in $Bl(\mathbb{L}(\mathbb{C}_t)) \cup Bl(\mathbb{L}(\mathbb{C}_{t+1}))$ and $Wh(\mathbb{L}(\mathbb{C}_t)) \cup Wh(\mathbb{L}(\mathbb{C}_{t+1}))$, respectively, appealing to part 2 of Proposition 7.5 again we get that $\max_{i\in[t,t+1]}\{\mathit{cost}(\mathbb{L}(\mathbb{C}_i))\} \geq \frac{1}{2}(\mathit{cost}(\mathbb{L}') - 3)$.

This establishes that even if the maximal cost in the L-pebbling $\mathcal{L}$ induced by derivation $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is attained in some intermediate L-configuration $\mathbb{L}' \notin \{\mathbb{L}(\mathbb{C}_t) \mid t \in [\tau]\}$, it still holds that $\max_{t\in[\tau]}\{\mathit{cost}(\mathbb{L}(\mathbb{C}_t))\} \geq \frac{1}{2}\mathit{cost}(\mathcal{L}) + \mathrm{O}(1)$. The theorem follows. ☐

9 A Separation of Space and Width in Resolution

In the last section, we proved that $Sp(Peb^d_{T_h} \vdash 0) = Sp({*}Peb^d_{T_h} \vdash \bigvee_{i=1}^d z_i)$, and that each resolution derivation $\pi : {*}Peb^d_{T_h} \to \bigvee_{i=1}^d z_i$ induces a legal L-pebbling $\mathcal{L}$ of T_h such that $\max_{\mathbb{C}\in\pi}\{\mathit{cost}(\mathbb{L}(\mathbb{C}))\} = \Omega(\mathit{cost}(\mathcal{L}))$. From Sections 6 and 7 we know that $\mathit{cost}(\mathcal{L}) = \Omega(\mathit{BW\text{-}Peb}(T))$. The final component needed to piece together the proof of our lower bound on the refutation space of pebbling contradictions is to show that the number of pebbles in an induced L-configuration $\mathbb{L}(\mathbb{C})$ and the number of clauses in $\mathbb{C}$ are somehow connected.

We cannot expect a proof of this fact to work regardless of the pebbling degree d. The induced L-pebbling in Section 8 makes no assumptions about d, but we know that $Sp({*}Peb^1_G \vdash z_1) = Sp(Peb^1_G \vdash 0) = \mathrm{O}(1)$. If we look at the resolution refutation π of Peb^1_G in constant space sketched at the end of Section 5, we see that the induced L-pebbling starts by placing white pebbles on $pred(z)$ and a black pebble on z, i.e., introducing $z\langle pred(z)\rangle$, and then pushes the white pebbles downwards by introducing $v\langle pred(v)\rangle$ for all v in reverse topological order and merging until it reaches $z\langle S\rangle$ for S the source vertices of G. Finally, the white pebbles $s \in S$ are eliminated one by one by introducing $s\langle\emptyset\rangle$ and merging. The reason that Peb^1_G can be refuted in constant space is that one single clause $z_1 \vee \bigvee_{v\in V} \overline{v}_1$ can induce an arbitrary number $|V|$ of white pebbles, or, phrasing it differently, that white pebbles are free for $d = 1$.

In Theorem 9.6 below, we prove a lower bound $|\mathbb{C}| \geq N$ for N induced pebbles. As we just observed, we will need $d \geq 2$ if some of these N pebbles are white. Black pebbles are *not* free for $d = 1$, however, but instead of showing a separate bound for them we assume $d \geq 2$ and give a simple, unified proof for

N simultaneous black or white pebbles. We conclude the section by combining the bound in Theorem 9.6 with previous theorems to obtain the tight bound on the refutation clause space of pebbling contradictions over binary trees in Theorem 1.1 and the separation of space and width in Corollary 1.2.

In the proofs, we will use material from Section 3 and the following definitions.

Definition 9.1. We say that a vertex v is *represented positively* in a clause C if $\{v_1, \dots, v_d\} \cap \mathit{Lit}(C) \neq \emptyset$ and *negatively* if $\{\overline{v}_1, \dots, \overline{v}_d\} \cap \mathit{Lit}(C) \neq \emptyset$, and that C *mentions* v *positively* or *negatively*, respectively. This definition is extended to sets of vertices and clauses by taking unions.

For a set of vertices U, we let $\mathit{Vars}^d(U) = \{u_1, \dots, u_d \mid u \in U\}$ denote the set of all variables representing vertices in U. For a set of clauses $\mathbb{C}$, we use $V(\mathbb{C}) = \{u \in U \mid \mathit{Vars}^d(u) \cap \mathit{Vars}(\mathbb{C}) \neq \emptyset\}$ to denote all vertices represented (positively or negatively) in $\mathbb{C}$, and we write $\mathbb{C}[U] = \{C \in \mathbb{C} \mid V(C) \cap U \neq \emptyset\}$ to denote the subset of all clauses in $\mathbb{C}$ mentioning vertices in U.

Definition 9.2. For v a vertex in T and α a truth value assignment, v is said to be *true under* α if $\alpha\left(\bigvee_{i=1}^d v_i\right) = 1$ and *false under* α if $\alpha\left(\bigvee_{i=1}^d v_i\right) = 0$. We define

$$\alpha^{v=\nu}(u_i) = \begin{cases} \alpha(u_i) & \text{if } u \neq v, \\ \nu & \text{if } u = v \end{cases}$$

and say that $\alpha^{v=0}$ *flips* v *to false*.

It is easy to see from Lemma 3.4 and Definition 8.6 that if a set of clauses $\mathbb{C}$ induces black pebbles on a set of vertices V, then these vertices must all be represented positively in $\mathbb{C}$. If $\mathbb{C}$ induces a white pebble on a vertex w, it follows immediately from Lemma 3.5 that all literals $\overline{w}_i$, $i \in [d]$, are present in $\mathit{Lit}(\mathbb{C})$. But we can say something stronger for white pebbles.

Lemma 9.3. *Suppose for a clause set $\mathbb{C}$ and a vertex w that there is a $v \in P_*^w$ and a $V \subseteq T \setminus P_*^w$ such that $\mathbb{C} \cup \mathbb{B}(V) \vDash A_{P^v}$ but $\mathbb{C} \cup \mathbb{B}(V \setminus \{w\}) \nvDash A_{P^v}$. Then there is a subset $\{\overline{w}_i \vee C_i \mid i \in [d]\} \subseteq \mathbb{C}$ for which $\overline{w}_j \notin \mathit{Lit}(C_i)$ if $j \neq i$.*

Proof. Pick α such that $\alpha(\mathbb{C}) = \alpha(\mathbb{B}(V \setminus \{w\})) = 1$ but $\alpha(A_{P^v}) = 0$. Then it must be the case that $\alpha\left(\bigvee_{i=1}^d w_i\right) = 0$. For all $i \in [d]$ we have $\alpha^{w_i=1}(\mathbb{B}(V)) = 1$ but $\alpha^{w_i=1}(A_{P^v}) = 0$, so flipping w_i while keeping w_j false for $j \neq i$ must falsify some clause in $\mathbb{C}$. This establishes that there are clauses $\overline{w}_i \vee C_i \in \mathbb{C}$ for all $i \in [d]$ such that $\overline{w}_j \notin \mathit{Lit}(C_i)$ for $j \neq i$. □

Lemma 9.3 tells us that one white pebble costs d clauses. We are convinced that the correct bound for N white pebbles should be dN clauses if $d \geq 2$.

We next prove a couple of lemmas to try to argue why the intuition for a bound dN, or at least $(d-1)N$, is strong. At the same time, the proofs of these lemmas indicate why such a bound appears hard to get. Loosely speaking, the problem seems to be that Theorem 3.6 does not really use any structural information about the CNF formula in question. Since very different formulas can yield the same clauses-variable occurrences bipartite graph, perhaps it should not be very surprising if the theorem does not always yield optimal bounds. The lemmas below are not used in the following, so the impatient reader is invited to skip ahead to Theorem 9.6 on page 45.

For N white pebbles in one common subconfiguration, the cost is at least $(d-1)N$ clauses.

Lemma 9.4. *If a clause set $\mathbb{C}$ derived from *Peb^d_T induces a subconfiguration $v\langle W\rangle$ then $|\mathbb{C}| > (d-1)|W|$.*

Proof. Pick $\mathbb{C}_v \subseteq \mathbb{C}$ and $V \subseteq T \setminus P^v$ minimal such that $W = \mathit{swp}(v, V)$ and $\mathbb{C}_v \cup \mathbb{B}(V) \vDash A_{P^v}$. Note that $\mathit{swp}(v, V) \subseteq V$ by definition. Since $\mathbb{B}(V)$ contains $d|V|$ variables but only $|V|$ clauses and $\mathit{Vars}(\mathbb{B}(V)) \cap \mathit{Vars}(A_{P^v}) = \emptyset$, Theorem 3.6 yields that $|\mathbb{C}| \geq |\mathbb{C}_v| > (d-1)|V| \geq (d-1)|W|$. ☐

Also, two white pebbles always cost at least $2d-1$ clauses, although here the argument starts to become pretty involved...

Lemma 9.5. *If a clause set $\mathbb{C}$ derived from *Peb^d_T induces two white pebbles on T, then $|\mathbb{C}| \geq 2d-1$.*

Proof. Suppose that $\mathbb{C}$ induces white pebbles on w^1 and w^2.

If w^1 and w^2 are contained in the same subconfiguration we have $|\mathbb{C}| \geq 2d-1$ by Lemma 9.4. Assume for $i = 1, 2$ that the induced subconfigurations are $v^i\langle W^i\rangle$, where $w^i \in W^i$, and let $\mathbb{C}_i \subseteq \mathbb{C}$ and $V^i \subseteq T \setminus P^{v^i}$ be minimal such that $W^i = \mathit{swp}(v^i, V^i)$ and $\mathbb{C}_i \cup \mathbb{B}(V^i) \vDash A_{P^{v^i}}$. If $|V^1| > 1$ or $|V^2| > 1$ we again have $|\mathbb{C}| \geq 2d-1$ by the proof of Lemma 9.4, so suppose that $V^i = W^i = \{w^i\}$.

Recall Definition 3.1 on page 7, and let $\rho_i = \rho(\neg A_{P^{v^i}})$ for $i = 1, 2$. Lemma 3.3 says that $(\mathbb{C}_i \cup \mathbb{B}(V^i))|_{\rho_i} = \mathbb{C}_i|_{\rho_i} \cup \mathbb{B}(V^i)$ is minimally unsatisfiable. Also, from the proof of Lemma 9.3 we can extract that $\mathbb{C}_i|_{\rho_i}$ contains d clauses $\{D^i_j = \overline{w}^i_j \vee C^i_j \mid j \in [d]\}$ for which $\overline{w}^i_k \notin \mathit{Lit}(C^i_j)$ if $k \neq j$. Let us refer to the literals $\overline{w}^i_j \in \mathit{Lit}(D^i_j)$ for $i = 1, 2$ and $j = 1, \ldots, d$ as *critical occurrences*. If $w^1 \in P^{v^2}$ we are done since ρ_2 kills all d clauses $\overline{w}^1_j \vee C^1_j$ and there are still d clauses $\overline{w}^2_j \vee C^2_j$ left in $\mathbb{C}_2|_{\rho_2}$, and the same holds for w^2 and P^{v^1} by symmetry. Assume therefore that $w^1 \notin P^{v^2}$ and $w^2 \notin P^{v^1}$.

Now if $|\mathbb{C}_1 \cup \mathbb{C}_2| < 2d$, Lemma 9.3 combined with the pigeonhole principle tells us that there is some negative literal, say $\overline{w}^1_1$, which occurs critically in a clause containing a literal $\overline{w}^2_j$. Consider the subset $\mathbb{C}_1[\{w^1, w^2\}]$ of clauses in $\mathbb{C}_1$ mentioning w^1 or w^2, and let $m = \mathit{Vars}(\mathbb{C}_1[\{w^1, w^2\}]) \cap \mathit{Vars}^d(w^2)$. We know that $\overline{w}^1_1$ occurs critically in $\mathbb{C}_1$ together with some $\overline{w}^2_j$, and that all literals from w^2 in $\mathbb{C}_1$ are present in $\mathbb{C}_1|_{\rho_1}$ as well, since $w^2 \notin P^{v^1}$ by assumption and ρ_1 does not satisfy any clauses in $\mathbb{C}_1$ by Lemma 3.3. Thus $m \geq 1$. By Theorem 3.6 we get $|\mathbb{C}_1[\{w^1, w^2\}]|_{\rho_1} \cup \mathbb{B}(w^1)| > d + m$, that is, $|\mathbb{C}_1[\{w^1, w^2\}]| \geq d + m$.

Since $\mathbb{C}_1[\{w^1, w^2\}] \subseteq \mathbb{C}_1 \cup \mathbb{C}_2$ and $|\mathbb{C}_1 \cup \mathbb{C}_2| < 2d$ we must have $m < d$. Consequently, there are $d - m \geq 1$ variables from w^2, say $w^2_1, \ldots, w^2_{d-m}$, that are not mentioned in $\mathbb{C}_1[\{w^1, w^2\}]$. But all negative literals $\overline{w}^i_j$ for $i = 1, 2$ and $j = 1, \ldots, d$ occur in $\mathbb{C}_1 \cup \mathbb{C}_2$, so the literals $\overline{w}^2_1, \ldots, \overline{w}^2_{d-m}$ can all be found in $(\mathbb{C}_1 \cup \mathbb{C}_2) \setminus \mathbb{C}_1[\{w^1, w^2\}]$. However,

$$\begin{aligned}
\left|(\mathbb{C}_1 \cup \mathbb{C}_2) \setminus \mathbb{C}_1[\{w^1, w^2\}]\right| &= |\mathbb{C}_1 \cup \mathbb{C}_2| - |\mathbb{C}_1[\{w^1, w^2\}]| \\
&\leq (2d-1) - (d+m) \\
&= d - (m+1),
\end{aligned}$$

which contradicts the existence of $d-m$ *distinct* clauses $\{\overline{w}^2_j \vee C^2_j \mid j \in [d-m]\}$ guaranteed by Lemma 9.3. Hence $|\mathbb{C}_1 \cup \mathbb{C}_2| \geq 2d$ and the lemma follows. ☐

We believe that the ideas in the proof of Lemma 9.5 could be pushed further to yield a bound $|\mathbb{C}| \geq (d-1)N$ for N white pebbles. However, to get a simpler proof, and to get a common bound for N simultaneous black *or* white pebbles, we instead opt for the bound $|\mathbb{C}| \geq N$.

Theorem 9.6. *Suppose that $\mathbb{C}$ is a set of clauses derived from $^*Peb^d_T$ for $d \geq 2$, and that $V \subseteq V(T)$ is a set of vertices such that $\mathbb{C}$ induces a black or white pebble on each $v \in V$, i.e., $V \subseteq Bl(\mathbb{L}(\mathbb{C})) \cup Wh(\mathbb{L}(\mathbb{C}))$. Then $|\mathbb{C}| \geq |V|$.*

Proof. Suppose that $\mathbb{C}$ induces a subconfiguration $v\langle W\rangle$. By Definition 8.6, there is a minimal support $V_v \subseteq T \setminus P^v$ such that $W = swp(v, V_v) \subseteq V_v$ and $\mathbb{C} \cup \mathbb{B}(V_v) \vDash A_{P^v}$ but $\mathbb{C} \cup \mathbb{B}(V_v) \nvDash A_{P^v_*}$ and $\mathbb{C} \cup \mathbb{B}(V'_v) \nvDash A_{P^v}$ for all $V'_v \subsetneqq V_v$.

Fix for each induced $v\langle W\rangle$ a subset $\mathbb{C}_v \subseteq \mathbb{C}$ such that $\mathbb{C}_v \cup \mathbb{B}(V_v) \vDash A_{P^v}$ minimally. Since $\mathbb{C}_v \cup \mathbb{B}(V_v) \nvDash A_{P^v_*}$ and $V_v \cap P^v = \emptyset$, the vertex v must be represented positively in $\mathbb{C}_v$ by Lemma 3.4. For the white pebbles in W, it follows from Lemma 9.3 (or even just from Lemma 3.5) that all literals $\{\overline{w}_i \mid w \in W, i \in [d]\}$ occur in $\mathbb{C}_v$.

We prove by induction over $U \subseteq V$ that $|\mathbb{C}[U]| \geq |U|$, from which the theorem clearly follows. The base case $|U| = 1$ is immediate, since we just proved that all pebbled vertices $v \in V$ are represented in $\mathbb{C}$.

For the induction step, suppose that $|\mathbb{C}[U']| \geq |U'|$ for all $U' \subsetneqq U$. Pick a "topmost" vertex $u \in U$, i.e., such that $P^u_* \cap U = \emptyset$, and look at the subconfiguration $v\langle W\rangle$ of u (with $v = u$ if u is black) and associated subset $\mathbb{C}_v \subseteq \mathbb{C}$. Note that $Vars^d(U) \cap Vars(A_{P^v}) \subseteq \{u\}$. Let $S = U \cap V(\mathbb{C}_v)$ be the set of all vertices in U mentioned by $\mathbb{C}_v$. We claim that $|\mathbb{C}_v[S]| \geq |S|$.

To show this, note first that it was proven above that $u \in S$, and if $S = \{u\}$ we trivially have $|\mathbb{C}_v[S]| \geq 1 = |S|$. Suppose therefore that $S \supsetneqq \{u\}$. We want to apply Theorem 3.6 on the formula $F = \mathbb{C}_v \cup \mathbb{B}(V_v)$. Let $S' = S \setminus \{u\}$, write $S' = S_1 \dot{\cup} S_2$ for $S_1 = S' \cap V_v$ and $S_2 = S' \setminus S_1$, and consider $F_{S'} = \{C \in (\mathbb{C}_v \cup \mathbb{B}(V_v)) \mid V(C) \cap S' \neq \emptyset\} = \mathbb{C}_v[S'] \cup \mathbb{B}(S_1)$. For each $w \in S_1$, the clauses in $\mathbb{B}(S_1)$ contain d literals $w_1, \ldots, w_d$, and these literals must all occur negated in $\mathbb{C}_v$ by Lemma 3.5. For each $w \in S_2$, the clauses in $\mathbb{C}_v[S']$ contain at least one variable w_i. Appealing to Theorem 3.6 with the subset of variables $Vars^d(S') \cap Vars(\mathbb{C}_v)$, we get that

$$|F_{S'}| = |\mathbb{C}_v[S'] \cup \mathbb{B}(S_1)| > |Vars^d(S') \cap Vars(\mathbb{C}_v)| \geq d|S_1| + |S_2|,$$

and rewriting this as

$$|\mathbb{C}_v[S]| \geq |\mathbb{C}_v[S']| = |F_{S'}| - |\mathbb{B}(S_1)| \geq (d-1)|S_1| + |S_2| + 1 \geq |S|$$

proves the claim.

Note that $\mathbb{C}_v[S] \subseteq \mathbb{C}[U]$, since $\mathbb{C}_v \subseteq \mathbb{C}$ and $S \subseteq U$. Also, by construction $\mathbb{C}_v[S]$ does not mention any vertices in $U \setminus S$ since $S = U \cap V(\mathbb{C}_v)$. In other words, $\mathbb{C}[U \setminus S] \subseteq \mathbb{C}[U] \setminus \mathbb{C}_v[S]$, and using the induction hypothesis for $U \setminus S \subsetneqq U$ we get

$$|\mathbb{C}[U]| \geq |\mathbb{C}_v[S]| + |\mathbb{C}[U \setminus S]| \geq |S| + |U \setminus S| = |U|.$$

The theorem follows by induction. ☐

We can now prove a tight bound for the refutation clause space of pebbling contradictions over binary trees.

Theorem 1.1 (restated). *Let T_h denote the complete binary tree of height h and $Peb^d_{T_h}$ the pebbling contradiction of degree $d \geq 2$ defined over T_h. Then the space of refuting $Peb^d_{T_h}$ by resolution is $Sp\left(Peb^d_{T_h} \vdash 0\right) = \Theta(h)$.*

Proof. The upper bound $Sp\left(Peb^d_G \vdash 0\right) \leq \mathit{Peb}(G) + \mathrm{O}(1)$ for any DAG G is fairly obvious: given an optimal black pebbling of G, derive $\bigvee_{i=1}^d v_i$ inductively when vertex v is pebbled. With a little care, this can be done in constant extra space independent of d. To see this, suppose for $pred(r) = \{p, q\}$ that a black pebble is placed on r. Then p and q are already black-pebbled, so we have $\bigvee_{i=1}^d p_i$ and $\bigvee_{j=1}^d q_j$ in memory. It is not hard to verify that $\overline{p}_i \vee \bigvee_{l=1}^d r_l$ can be derived in additional space 3 by resolving $\bigvee_{j=1}^d q_j$ with $\overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^d r_l$ for $j \in [d]$. Resolve $\bigvee_{i=1}^d p_i$ with $\overline{p}_1 \vee \bigvee_{l=1}^d r_l$ to get $\bigvee_{i=2}^d p_i \vee \bigvee_{l=1}^d r_l$, and then resolve this clause with $\overline{p}_i \vee \bigvee_{l=1}^d r_l$ for $i = 2, \ldots, d$ to get $\bigvee_{l=1}^d r_l$ in total extra space 4. Conclude the resolution proof by resolving $\bigvee_{i=1}^d z_i$ for the target z with the target axioms $\overline{z}_i$, $i \in [d]$, in space 3. Consequently, $Sp\left(Peb^d_{T_h} \vdash 0\right) = \mathrm{O}\big(\mathit{Peb}(T_h)\big) = \mathrm{O}(h)$.

For the lower bound, according to Observation 8.2 we have $Sp\left(Peb^d_G \vdash 0\right) = Sp\left({*Peb}^d_G \vdash \bigvee_{i=1}^d z_i\right)$. Let $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ be a resolution derivation of $\bigvee_{i=1}^d z_i$ from ${*Peb}^d_{T_h}$ in minimal clause space. Combining Theorems 4.3, 6.11 and 8.13, we know that the derivation π induces a legal L-pebbling $\mathcal{L}$ of the tree T_h such that there is a clause configuration $\mathbb{C}_t \in \pi$ with $\mathit{cost}(\mathbb{L}(\mathbb{C}_t)) = \Omega\big(\mathit{cost}(\mathcal{L})\big) = \Omega(\mathit{BW\text{-}Peb}(T_h)) = \Omega(h)$. Fix such a clause configuration $\mathbb{C}_t$. By Theorem 9.6, $\left|\mathbb{C}_t\right| \geq \left|Bl\big(\mathbb{L}(\mathbb{C}_t)\big) \cup Wh\big(\mathbb{L}(\mathbb{C}_t)\big)\right| = \mathit{cost}(\mathbb{L}(\mathbb{C}_t)) = \Omega(h)$.

It follows that $Sp\left(Peb^d_{T_h} \vdash 0\right) = \Theta(h)$ for $d \geq 2$. ☐

Since $W\big(Peb^d_G \vdash 0\big) = \mathrm{O}(d)$ for all pebbling contradictions by Theorem 5.2, fixing $d \geq 2$ in Theorem 1.1 yields a separation of clause space from width. Corollary 1.2 follows if we let $F_n = Peb^d_{T_h}$ for $h = \lfloor \log(n+1) \rfloor$.

Corollary 1.2 (restated). *For all $k \geq 4$, there is a family of k-CNF formulas $\left\{F_n\right\}_{n=1}^\infty$ of size $\mathrm{O}(n)$ such that $W(F_n \vdash 0) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Theta(\log n)$.*

10 Conclusion and Open Problems

We have proven an asymptotically tight bound on the refutation clause space in resolution of pebbling contradictions over binary trees. Our result is the first lower bound on refutation space which is not the consequence of a lower bound on the refutation width for the same formulas, but instead separates the two measures.

This answers an open question in [9, 23, 25, 39]. However, we believe that our answer can be strengthened in several ways.

Firstly, we would like to extend the lower bound on the refutation space of pebbling contradictions over binary trees to the k-DNF resolution proof systems $\mathfrak{Res}(k)$ introduced in [31], where the configurations $\mathbb{C}$ consist of k-DNF formulas instead of CNF clauses and one can "resolve" over up to k variables simultaneously. It is easy to prove the generalization of Theorem 5.3 that $Sp_{\mathfrak{Res}(k)}\big(Peb^d_G \vdash 0\big) = \mathrm{O}(1)$ if $d \leq k$. We believe that pebbling contradictions $Peb^{k+1}_{T_h}$ separate k-DNF resolution and $(k+1)$-DNF resolution with respect to space.

Conjecture 10.1. *For k-DNF resolution refutations of pebbling contradictions on complete binary trees, fixing k it holds that $W_{\mathfrak{Res}(k)}\bigl(Peb^{k+1}_{T_h} \vdash 0\bigr) = \mathrm{O}(1)$ and $Sp_{\mathfrak{Res}(k)}\bigl(Peb^{k+1}_{T_h} \vdash 0\bigr) = \Omega(h)$ but $Sp_{\mathfrak{Res}(k+1)}\bigl(Peb^{k+1}_{T_h} \vdash 0\bigr) = \mathrm{O}(1)$.*

Proving this conjecture would establish that the k-DNF resolution proof systems form a strict hierarchy with respect to space, which would be an improvement of the separation result in [23] for the restricted case of tree-like k-DNF resolution.

Secondly, it would be nice to generalize the bound on refutation space of pebbling contradictions to DAGs other than trees that have better size-pebbling price trade-off. For $L = L(F)$ the number of clauses in a formula F, we want to find a formula family which improves the the bound $Sp\bigl(F \vdash 0\bigr) = \Omega\bigl(\log L\bigr)$ in Theorem 1.1 to $Sp\bigl(F \vdash 0\bigr) = \Omega\bigl(L\bigr)$ or at least $Sp\bigl(F \vdash 0\bigr) = \Omega\bigl(L^{\epsilon}\bigr)$ for some constant $\epsilon > 0$, but for which it still holds that $W\bigl(F \vdash 0\bigr) = \mathrm{O}(1)$.

We conjecture that the black-white pebbling price is a lower bound for pebbling contradictions over any DAG.

Conjecture 10.2. *For $d \geq 2$ and for G an arbitrary DAG with a unique target and with all vertices having indegree 0 or 2, $Sp\bigl(Peb^{d}_{G} \vdash 0\bigr) = \Omega\bigl(\textsf{BW-Peb}(G)\bigr)$.*

Since there are DAGs G_n of fan-in 2 and size $\mathrm{O}(n)$ which have black-white pebbling price $\textsf{BW-Peb}(G_n) = \Theta\bigl(n/\log n\bigr)$ (see [27]),[4] a proof of Conjecture 10.2 would immediately yield the following corollary.

Corollary 10.3 (assuming Conjecture 10.2). *There is a family of unsatisfiable k-CNF formulas $\bigl\{F_n\bigr\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ such that $W\bigl(F_n \vdash 0\bigr) = \mathrm{O}(1)$ but $Sp\bigl(F_n \vdash 0\bigr) = \Omega\bigl(n/\log n\bigr)$.*

A third and final question is whether refutation space can be separated from refutation length in the sense that there can be shown to exist a polynomial-size family of k-CNF formulas such that $Sp\bigl(F \vdash 0\bigr) = \omega\bigl(\sqrt{n \log L(F \vdash 0)}\bigr)$, where n is the number of variables in F. This would be an interesting contrast to the relation $W\bigl(F \vdash 0\bigr) = \mathrm{O}\bigl(\sqrt{n \log L(F \vdash 0)}\bigr)$ between length and width proven in [12]. We believe that such a formula family exists.

Conjecture 10.4. *There is a family of k-CNF formulas $\bigl\{F_n\bigr\}_{n=1}^{\infty}$ over n variables such that $Sp\bigl(F_n \vdash 0\bigr) = \omega\bigl(\sqrt{n \log L(F_n \vdash 0)}\bigr)$.*

Of course, if we could prove Conjecture 10.2, we would immediately get a positive answer to Conjecture 10.4 as well, using the same formula family as in Corollary 10.3.

It is not possible to prove Conjecture 10.2 by using the L-pebble game of Section 6 on general DAGs G, though. As was observed in Section 6, if we allow reversal moves of black pebbles downwards it is not true that $\textsf{L-Peb}(G) = \Omega\bigl(\textsf{BW-Peb}(G)\bigr)$.

As a first step, we would therefore have to modify Definition 8.6 so that a set of clauses $\mathbb{C}$ induces a black pebble on v if there is a minimal subset $\mathbb{C}_v \subseteq \mathbb{C}$ such that $\mathbb{C}_v \cup \mathbb{B}\bigl(V\bigr) \vDash A_{P^v}$ but $\mathbb{C}_v \cup \mathbb{B}\bigl(V\bigr) \nvDash A_{P^v_*}$. Otherwise we could move black pebbles downwards through erasures simply by deriving

[4] Note that in several papers, this result is incorrectly attributed to [33], but [33] itself gives the correct reference.

$\mathbb{C} = \{\bigvee_{i=1}^{d} v_i,\ \bigvee_{i=1}^{d} succ(v)_i\}$ and then deleting $\bigvee_{i=1}^{d} succ(v)_i$. But if we define induced pebbles in terms of subsets $\mathbb{C}_v \subseteq \mathbb{C}$, as a result black pebbles can slide downwards after inference steps, since $\{B \vee C\}$ is weaker than $\{B \vee x,\ C \vee \overline{x}\}$. This problem can be solved by defining induced pebbles syntactically instead of semantically. Recalling Definition 3.7 on page 9, we could say that $\mathbb{C}$ induces a black pebble on v if there is a $\mathbb{C}_v \subseteq \mathbb{C}$ with v represented positively in $\mathbb{C}_v$ such that $\mathbb{C}_v \cup \mathbb{B}(V) \vDash_{\triangledown} D_v \subseteq A_{P^v}$. With this definition, nothing bad happens during inference or erasure steps, resolution derivations yield legal L-pebblings, and Theorem 3.10 can be used to show the bound in Theorem 1.1. However, because of the fact that the support $\mathbb{B}(w) = \bigvee_{l=1}^{d} w_l$ for a non-leaf w is stronger than $Ax^d(w) = \{\overline{u}_i \vee \overline{v}_j \vee \bigvee_{l=1}^{d} w_l \mid i, j \in [d]\}$, we can still get black pebbles moving downwards at axiom download. This can be avoided by defining support in terms of $Ax^d(w)$ instead of $\bigvee_{i=1}^{d} w_i$, which leads to a very nice pebble game, but then unfortunately the counting argument in Theorem 3.10 to get a bound on $|\mathbb{C}|$ in terms of the number of induced pebbles breaks down.

These problems arise because we do not a priori have any restrictions on what kind of clauses a resolution derivation from a pebbling contradiction might derive. The counterexample derivations we have found for the definitions sketched in the previous paragraph all seem clearly non-optimal, while all of the definitions yield well-behaved pebblings for "normal" resolution derivations. One way of solving the problems would be if one could define formally what constitutes a "non-optimal" derivation from a pebbling contradiction and then show that each non-optimal derivation can be replaced by an "optimal" one in at most the same space. Alternatively, one could try to find new ideas for the connection between the black-white pebble game and resolution derivations from pebbling contradictions, or use the last definition for induced pebbles outlined above but devise new methods for proving bounds on $|\mathbb{C}|$ in terms of the number of induced pebbles.

Acknowledgements

I am grateful to my supervisor, Professor Johan Håstad, for patiently giving feed-back during this work. Also, I would like to thank Douglas Wikström for interesting discussions about theoretical computer science in general and pebble games in particular. Finally, a special thanks to Joel Brynielsson, who introduced me to the mysteries of METAPOST.

References

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

[2] 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.

[3] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolu-

tion. In *Proceedings 34th Annual ACM Symposium on Theory of Computing (STOC '02)*, pages 448–456, May 2002.

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

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

[6] Paul Beame. *Proof Complexity*. Lecture notes, Institute for Advanced Study, 2000.

[7] 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.

[8] Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. *Bulletin of the European Association for Theoretical Computer Science*, 65:66–89, June 1998.

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

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

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

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

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

[14] 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.

[15] Maria Luisa Bonet and Nicola Galesi. A study of proof search algorithms for resolution and polynomial calculus. In *Proceedings 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS '99)*, pages 422–431, October 1999.

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

[17] Samuel R. Buss, editor. *Handbook of Proof Theory*. Elsevier Science, Amsterdam, 1998.

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

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

[20] 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.

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

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

[23] 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.

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

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

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

[27] 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 `http://www-db.stanford.edu/TR/CS-TR-78-661.html`.

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

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

[30] Hans Kleine Büning and Theodor Letterman. *Propositional Logic: Deduction and Algorithms.* Cambridge University Press, 1999.

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

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

[33] Thomas Lengauer and Robert Endre Tarjan. Upper and lower bounds on time-space tradeoffs. In *Proceedings 11th Annual ACM Symposium on Theory of Computing (STOC '79)*, pages 262–277, May 1979.

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

[35] 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.

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

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

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

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

[40] 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.

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

[42] Alasdair Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 1(4):425–467, 1995.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'