THEORY OF COMPUTING, Volume 9 (14), 2013, pp. 471–557
www.theoryofcomputing.org

Towards an Optimal Separation of Space and Length in Resolution*

Jakob Nordström† Johan Håstad‡

Received February 16, 2011; Revised March 9, 2013; Published May 27, 2013

Abstract: Most state-of-the-art satisfiability algorithms today are variants of the DPLL procedure augmented with clause learning. The main bottleneck for such algorithms, other than the obvious one of time, is the amount of memory used. In the field of proof complexity, the resources of time and memory correspond to the length and space of resolution proofs. There has been a long line of research trying to understand these proof complexity measures, as well as relating them to the width of proofs, i.e., the size of a largest clause in the proof, which has been shown to be intimately connected with both length and space. While strong results have been proven for length and width, our understanding of space has been quite poor. For instance, it has remained open whether the fact that a formula is provable in short length implies that it is also provable in small space (which is the case for length versus width), or

*A preliminary version of this paper appeared in the *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08).*

†Part of this work done while at the Massachusetts Institute of Technology supported by the Royal Swedish Academy of Sciences, the Ericsson Research Foundation, the Sweden-America Foundation, the Foundation Olle Engkvist Byggmästare, and the Foundation Blanceflor Boncompagni-Ludovisi, née Bildt. Currently supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no 279611 and by Swedish Research Council grants 621-2010-4797 and 621-2012-5645.

‡Supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) / ERC grant agreement no 226203.

ACM Classification: F.4.1, F.1.3, I.2.3

AMS Classification: 68Q05, 68Q15, 68Q17, 68T15

Key words and phrases: proof complexity, resolution, length, space, width, separation, lower bound, pebbling, black-white pebble game

 DOI: 10.4086/toc.2013.v009a014

whether these measures are unrelated in the sense that short proofs can be arbitrarily complex with respect to space.

In this paper, we present some evidence indicating that the latter case should hold and provide a roadmap for how such an optimal separation result could be obtained. We do so by proving a tight bound of $\Theta(\sqrt{n})$ on the space needed for so-called pebbling contradictions over pyramid graphs of size n. This yields the first polynomial lower bound on space that is not a consequence of a corresponding lower bound on width, as well as an improvement of the weak separation of space and width (Nordström, STOC 2006) from logarithmic to polynomial.

Contents

1 Introduction **474**
- 1.1 Previous work . . . 475
- 1.2 Questions left open by previous research . . . 476
- 1.3 Our contribution . . . 477
- 1.4 Subsequent developments . . . 478

2 Proof overview and paper organization **478**
- 2.1 Sketch of preliminaries . . . 478
- 2.2 Proof idea for pebbling contradictions space bound . . . 479
- 2.3 Detailed overview of formal proof of space bound . . . 481
- 2.4 Paper organization . . . 484

3 Formal preliminaries **484**
- 3.1 The resolution proof system . . . 484
- 3.2 Pebble games and pebbling contradictions . . . 486

4 A game for analyzing pebbling contradictions **488**
- 4.1 Some graph notation and definitions . . . 488
- 4.2 Description of the blob-pebble game and formal definition . . . 489
- 4.3 Blob-Pebbling price . . . 493

5 Resolution derivations induce blob-pebblings **495**
- 5.1 Definition of induced configurations and theorem statement . . . 495
- 5.2 Some technical lemmas . . . 497
- 5.3 Erasure . . . 498
- 5.4 Inference . . . 498
- 5.5 Axiom download . . . 498
- 5.6 Wrapping up the proof of Theorem 5.3 . . . 502

6 Induced blob configurations measure clause set size **503**

7 Black-white pebbling and layered graphs **508**
- 7.1 Some preliminaries and a tight bound for black pebbling . . . 508
- 7.2 A tight bound on the black-white pebbling price of pyramids . . . 511
- 7.3 An exposition of the proof of the limited hiding-cardinality property . . . 516

8 A tight bound for blob-pebbling the pyramid **526**
- 8.1 Definitions and notation for the blob-pebbling price lower bound . . . 527
- 8.2 A lower bound assuming an extension of the LHC property . . . 529
- 8.3 Some structural transformations . . . 531
- 8.4 Proof of the generalized limited hiding-cardinality property . . . 534
- 8.5 Recapitulation of the proof of Theorem 1.1 and optimality of result . . . 547

9 Conclusion and open problems **548**

1 Introduction

Ever since the fundamental NP-completeness result of Cook [24], the problem of deciding whether a given propositional logic formula in conjunctive normal form (CNF) is satisfiable or not has been on center stage in Theoretical Computer Science. In more recent years, SATISFIABILITY has gone from a problem of mainly theoretical interest to a practical approach for solving applied problems. Although all known Boolean satisfiability solvers (SAT solvers) have exponential running time in the worst case, enormous progress in performance has led to satisfiability algorithms becoming a standard tool for solving a large number of real-world problems such as hardware and software verification, experiment design, circuit diagnosis, and scheduling.

A rather surprising aspect of this development is that the most successful SAT solvers to date are still variants of the resolution-based Davis-Putnam-Logemann-Loveland (DPLL) procedure [28, 29] augmented with *clause learning* [7, 46]. For instance, the great majority of the best algorithms in recent rounds of the international SAT competitions [58] fit this description. DPLL procedures perform a recursive backtrack search in the space of partial truth value assignments. The idea behind clause learning is that at each failure (backtrack) point in the search tree, the system derives a reason for the inconsistency in the form of a new clause and then adds this clause to the original CNF formula ("learning" the clause). This can save a lot of work later on in the proof search, when some other partial truth value assignment fails for similar reasons. The second main bottleneck for this approach, in addition to the obvious one of time, is the amount of memory used by the algorithms. Since there is only a limited amount of space, all clauses cannot be stored. The difficulty lies in obtaining a highly selective and efficient clause caching scheme that nevertheless keeps the clauses needed. Thus, understanding time and memory requirements for clause learning algorithms, and how these requirements are related to one another, is a question of great practical importance. We refer to, e.g., [41, 45] for a more detailed discussion SAT solving with examples of applications.

The study of proof complexity originated with the seminal paper of Cook and Reckhow [26]. In its most general form, a proof system for a language L is a predicate $P(x,\pi)$, computable in time polynomial in $|x|$ and $|\pi|$, such that for all $x \in L$ there is a string π (a *proof*) for which $P(x,\pi) = 1$, whereas for any $x \notin L$ it holds for all strings π that $P(x,\pi) = 0$. A proof system is said to be polynomially bounded if for every $x \in L$ there is a proof π_x of size at most polynomial in $|x|$. A *propositional proof system* is a proof system for the language of tautologies in propositional logic.

From a theoretical point of view, one important motivation for proof complexity is the intimate connection with the question of P versus NP. 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 [26] that NP = co-NP if and only if there exists a polynomially bounded propositional proof system. Hence, if it could be shown that there are no such proof systems, $\mathsf{P} \neq \mathsf{NP}$ would follow as 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, although great progress has been made in the last couple of decades for a variety of proof systems, it seems that we are still very far from fully understanding the reasoning power of even quite simple ones.

A second important motivation is that, as was mentioned above, designing efficient algorithms for proving tautologies (or, equivalently, testing satisfiability), is an important problem not only in the theory of computation but also in applied research and industry. 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. Proof complexity analyzes what it takes to simply write down and verify the proofs that such an automated theorem prover might find, ignoring the computational effort needed to actually find them. Thus, a lower bound for a proof system tells us that any algorithm, even an optimal (non-deterministic) one making all the right choices, must necessarily use at least the amount of a certain resource specified by this bound. In the other direction, theoretical upper bounds on some proof complexity measure give us hope of finding good proof search algorithms with respect to this measure, provided that we can design algorithms that search for proofs in the system in an efficient manner. For DPLL procedures with clause learning, the time and memory resources used are measured by the *length* and *space* of proofs in the resolution proof system.

The field of proof complexity also has rich connections to cryptography, artificial intelligence and mathematical logic. Some good surveys providing more details are [8, 11, 59].

1.1 Previous work

Any formula in propositional logic can be converted to a CNF formula that is only linearly larger and is unsatisfiable if and only if the original formula is a tautology. Therefore, any sound and complete system for refuting CNF formulas can be considered as a general propositional proof system.

Perhaps the single most studied proof system in propositional proof complexity, *resolution*, is such a system that produces proofs of the unsatisfiability of CNF formulas. The resolution proof system appeared in [18] and began to be investigated in connection with automated theorem proving in the 1960s [28, 29, 56]. Because of its simplicity—there is only one derivation rule—and because all lines in a proof are clauses, this proof system readily lends itself to proof search algorithms.

Being so simple and fundamental, resolution was also a natural target to attack when developing methods for proving lower bounds in proof complexity. In this context, it is more convenient to prove bounds on the *length* of refutations, i. e., the number of clauses, rather than on the total size of refutations. The length and size measure differ by at most a multiplicative factor given by the number of variables and are hence polynomially related. In 1968, Tseitin [61] presented a superpolynomial lower bound on refutation length for a restricted form of resolution, called *regular* resolution, but it was not until almost 20 years later that Haken [36] proved the first superpolynomial lower bound for general resolution. This (weakly) exponential lower bound of Haken has later been followed by many other strong results on resolution refutation length for different formula families, e. g., in [10, 17, 22, 23, 52, 54, 55, 62].

A second complexity measure for resolution, first made explicit by Galil [33], is the *width*, measured as the maximal size of a clause in the refutation. Ben-Sasson and Wigderson [17] showed that the minimal width $W(F \vdash \bot)$ of any resolution refutation of a k-CNF formula F is bounded from above by the minimal refutation length $L(F \vdash \bot)$ by

$$W(F \vdash \bot) = \mathrm{O}\left(\sqrt{n \log L(F \vdash \bot)}\right), \tag{1.1}$$

where n is the number of variables in F. Since it is also easy to see that refutations of polynomial-size formulas in small width must necessarily be short (simply for the reason that $(2 \cdot \#\text{variables})^w$ is an upper bound on the total number of distinct clauses of width w), the result in [17] can be interpreted as saying roughly that there exists a short refutation of the k-CNF formula F if and only if there exists a (reasonably) narrow refutation of F. This gives rise to a natural proof search heuristic: to find a short refutation, search for refutations in small width. It was shown in [14] that there are formula families for which this heuristic exponentially outperforms any DPLL procedure regardless of branching function.

The formal study of *space*[1] in resolution was initiated by Esteban and Torán [31]. Intuitively, the space $Sp(\pi)$ of a refutation π is the maximal number of clauses one needs to keep in memory while verifying the refutation, and the space $Sp(F \vdash \bot)$ of refuting F is defined as the minimal space of any resolution refutation of F. A number of upper and lower bounds for refutation space in resolution and other proof systems were subsequently presented in, for example, [2, 13, 30, 32]. Just as for width, the minimum space of refuting a formula can be upper-bounded by the size of the formula. Somewhat unexpectedly, however, it also turned out that the lower bounds on refutation space for several different formula families exactly matched previously known lower bounds on refutation width. Atserias and Dalmau [5] showed that this was not a coincidence, but that the inequality

$$W(F \vdash \bot) \le Sp(F \vdash \bot) + \mathrm{O}(1) \tag{1.2}$$

holds for any k-CNF formula F, where the (small) constant term depends on k. In [47], the first author proved that the inequality (1.2) is asymptotically strict by exhibiting a k-CNF formula family of size $\mathrm{O}(n)$ refutable in width $W(F_n \vdash \bot) = \mathrm{O}(1)$ but requiring space $Sp(F_n \vdash \bot) = \Theta(\log n)$.

1.2 Questions left open by previous research

Despite all the research that has gone into understanding the resolution proof system, a number of fundamental questions have remained unsolved. We touch briefly on two such questions below, and then discuss a third one, which is the main focus of this paper, in somewhat more detail.

As was mentioned above, equation (1.1) says that short refutation length implies narrow refutation width. Observe, however, that this does *not* mean that there is a refutation that is both short and narrow, since there is no guarantee that the refutations on the left- and right-hand sides of (1.1) are the same one. An intriguing open question is whether small length and width can always be achieved simultaneously, or whether there is a trade-off between these two measures. For the restricted case of so-called *tree-like* resolution it is known that there can be strong trade-offs [12], but the case of the (much more powerful) general resolution proof system has remained open.

A second, analogous problem concerns space and length. Combining equation (1.2) with the observation above that narrow refutations are trivially short, one can immediately conclude that small refutation clause space implies short refutation length. But again, this does *not* imply that any small-space refutation must also be short. In fact, it was shown in [12] that the refutations on the two sides of the inequality (1.2) in general cannot be the same one. An interesting question is whether small space of a refutation implies that it can also be made short, or whether space and length might have to be traded off against one another.

[1] The space measure discussed in this introduction is known as *clause space*. Another natural, but less studied, space measure is *total space*, which counts the maximal number of variable occurrences that must be kept in memory simultaneously. The focus of the current paper, however, is almost exclusively on clause space.

A third, even more fundamental question is whether short length has any implications for space. Note that for width, rewriting the bound in (1.1) in terms of the number of clauses $|F_n|$ instead of the number of variables tells us that if the width of refuting F_n is $\omega\big(\sqrt{|F_n|\log|F_n|}\big)$, then the length of refuting F_n must be superpolynomial in $|F_n|$. This is known to be almost tight, since [20] shows that there is a k-CNF formula family $\{F_n\}_{n=1}^{\infty}$ that requires width $\Omega\big(\sqrt[3]{|F_n|}\big)$ but nevertheless can be refuted in length $\mathrm{O}(|F_n|)$. Hence, formula families refutable in polynomial length can have somewhat wide minimum-width refutations, but not arbitrarily wide ones.

What does the corresponding relation between length and space look like? The inequality (1.2) tells us that any correlation between length and clause space cannot be tighter than the correlation between length and width, so in particular we get from the previous paragraph that k-CNF formulas refutable in polynomial length may have at least "somewhat spacious" minimum-space refutations. At the other end of the spectrum, given any resolution refutation π of F in length L it can be proven using results from [31, 39] that the space needed is at most $\mathrm{O}(L/\log L)$. This gives an upper bound on any possible separation of the two measures. But is there a Ben-Sasson–Wigderson style upper bound on space in terms of length similar to (1.1)? Or are length and space on the contrary unrelated in the sense that there exist k-CNF formulas F_n with short refutations but maximal possible refutation space $Sp(F_n \vdash \perp) = \Omega\big(L(F_n \vdash \perp)/\log L(F_n \vdash \perp)\big)$ in terms of length?

We remark that for tree-like resolution, [31] showed that there is a tight correspondence between length and space, exactly as for length versus width. The case for general resolution has been discussed in, e. g., [12, 32, 60], but there has been no consensus on what the right answer should be. However, these papers identify a plausible formula family for answering the question, namely so-called *pebbling contradictions* defined in terms of pebble games over directed acyclic graphs.

1.3 Our contribution

The main result in this paper provides an indication that the true answer to the question about the relationship between space and length is more likely to be at the latter extreme, i. e., that the two measures can be separated in the strongest sense possible. More specifically, as a step towards reaching this goal we prove an asymptotically tight bound on the clause space of refuting pebbling contradictions over so-called pyramid graphs.

Theorem 1.1. *The clause space of refuting pebbling contradictions over pyramid graphs of height h in resolution grows as $\Theta(h)$, provided that the number of variables per vertex in the pebbling contradictions is at least 2.*

This theorem yields the first separation of space and length (in the sense of a polynomial lower bound on space for formulas refutable in polynomial length) that is not a consequence of a corresponding lower bound on width, as well as an exponential improvement of the separation of space and width in [47]. Namely, from Theorem 1.1 we easily obtain the following corollary.

Corollary 1.2. *For all $k \geq 4$, there is a family $\{F_n\}_{n=1}^{\infty}$ of k-CNF formulas of size $\Theta(n)$ that can be refuted in resolution in length $L(F_n \vdash \perp) = \mathrm{O}(n)$ and width $W(F_n \vdash \perp) = \mathrm{O}(1)$ but require clause space $Sp(F_n \vdash \perp) = \Theta(\sqrt{n})$.*

1.4 Subsequent developments

In a joint paper [15] by Ben-Sasson and the first author, the separation in Corollary 1.2 has been improved to a clause space lower bound $Sp(F_n \vdash \perp) = \Omega(n/\log n)$ while still keeping the upper bounds on length $L(F_n \vdash \perp) = \mathrm{O}(n)$ and width $W(F_n \vdash \perp) = \mathrm{O}(1)$. This is essentially optimal up to multiplicative constants (except possibly for a logarithmic factor in the space-width separation). The construction in [15] follows the general roadmap laid out in the current paper, but changes the family of formulas under consideration in Theorem 1.1. The new results are therefore incomparable with those in the current paper in that the techniques used in [15] cannot prove Theorem 1.1, whereas our techniques, although similar, do not extend to the results in [15].

Even considering the progress made in [15], we believe that the results presented in our paper retain independent interest. This is so since our formula families are simpler, and an improvement of our techniques could conceivably yield optimal, tight results up to constant *additive* terms. This, in turn, could possibly be used to settle the question how hard it is to decide the space requirements for refuting a k-CNF formula. This problem is easily seen to be in PSPACE but is not known to be PSPACE-complete. Due to the inherent space blow-up between upper and lower bounds in [15], it is hard to envision the results from there being used for similar purposes. We elaborate briefly on this issue in Section 9.

2 Proof overview and paper organization

Since the proof of our main theorem is fairly involved, we start by giving an intuitive, high-level description of the proofs of our results and outlining how this paper is organized.

2.1 Sketch of preliminaries

A *resolution refutation* of a CNF formula F can be viewed as a sequence of derivation steps on a blackboard. In each step we may write a clause from F on the blackboard (an *axiom* clause), erase a clause from the blackboard or derive some new clause implied by the clauses currently written on the blackboard.[2] The refutation ends when we reach the contradictory empty clause. The *length* of a resolution refutation is the number of clauses in the refutation, the *width* is the size of the largest clause in the refutation, and the *clause space* is the maximum number of clauses on the blackboard simultaneously. We write $L(F \vdash \perp)$, $W(F \vdash \perp)$ and $Sp(F \vdash \perp)$ to denote the minimum length, width and clause space, respectively, of any resolution refutation of F.

The *pebble game* played on a directed acyclic graph (DAG) G models the calculation described by G, where the source vertices contain the input and non-source vertices specify operations on the values of the predecessors (see Figure 1). Placing a pebble on a vertex v corresponds to storing in memory the partial result of the calculation described by the subgraph rooted at v. Removing a pebble from v corresponds to deleting the partial result of v from memory. A *pebbling* of a DAG G is a sequence of moves starting with an empty graph G without pebbles and ending with all vertices in G empty except for a pebble on the (unique) sink vertex. The *cost* of a pebbling is the maximal number of pebbles used simultaneously

[2] For our proof, it turns out that the exact definition of the derivation rule is not essential—our lower bound holds for any sound rule. What is important is that we are only allowed to derive new clauses from the set of clauses currently on the board.

(a) DAG encoding calculation. (b) After pebbling with results filled in.

Figure 1: Example of modelling calculation as pebbling of DAG.

at any point in time during the pebbling. The *pebbling price* of a DAG G is the minimum cost of any pebbling, i. e., the minimum number of memory registers required to perform the complete calculation described by G.

The pebble game on a DAG G can be encoded as an unsatisfiable CNF formula Peb^d_G, a so-called *pebbling contradiction* of degree d. See Figure 2 for a small example. Very briefly, pebbling contradictions are constructed as follows:

- Associate d variables $x(v)_1, \dots, x(v)_d$ with each vertex v (in Figure 2 we have $d = 2$).
- Specify that all sources have at least one true variable, for example, the clause $x(r)_1 \lor x(r)_2$ for the vertex r in Figure 2.
- Add clauses saying that truth propagates from predecessors to successors. For instance, for the vertex u with predecessors r and s, clauses 4–7 in Figure 2 are the CNF encoding of the implication $(x(r)_1 \lor x(r)_2) \land (x(s)_1 \lor x(s)_2) \to (x(u)_1 \lor x(u)_2)$.
- To get a contradiction, conclude the formula with $\overline{x(z)}_1 \land \cdots \land \overline{x(z)}_d$ where z is the sink of the DAG.

We will need the observation from [14] that a pebbling contradiction of degree d over a graph with n vertices can be refuted by resolution in length $\mathrm{O}\big(d^2 \cdot n\big)$ and width $\mathrm{O}(d)$.

2.2 Proof idea for pebbling contradictions space bound

Pebble games have been used extensively as a tool to prove time and space lower bounds and trade-offs for computation. Loosely put, a lower bound for the pebbling price of a graph says that although the computation that the graph describes can be performed quickly (any graph can be pebbled in linear time), it requires large space. Our hope is that when we encode pebble games in terms of CNF formulas, these formulas inherit the same properties as the underlying graphs. That is, if we pick a DAG G with high pebbling price, since the corresponding pebbling contradiction encodes a calculation which requires large memory we would like to try to argue that any resolution refutation of this formula should require large space. Then a separation result would follow since we already know from [14] that the formula can be refuted in short length.

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

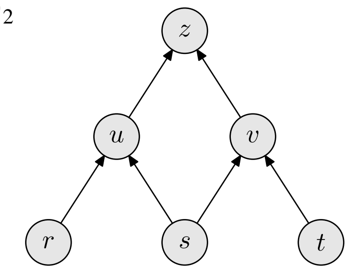


Figure 2: The pebbling contradiction $Peb^2_{\Pi_2}$ for the pyramid graph Π_2 of height 2.

More specifically, what we would like to do is to establish a connection between resolution refutations of pebbling contradictions on the one hand, and the so-called *black-white pebble game* [27] modelling the non-deterministic computations described by the underlying graphs on the other. Our intuition is that the resolution proof system should have to conform to the combinatorics of the pebble game in the sense that from any resolution refutation of a pebbling contradiction Peb^d_G we should be able to extract a pebbling of the DAG G. Our goal is to prove a lower bound on the resolution refutation space of pebbling contradictions reasoning along the following lines:

1. First, find a way of interpreting sets of clauses currently "on the blackboard" in a refutation of the formula Peb^d_G in terms of black and white pebbles on the vertices of the DAG G.

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

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

Combining the above with known lower bounds on the pebbling price of G, this would imply a lower bound on the refutation space of pebbling contradictions and a separation from length and width. For clarity, let us spell out what the formal argument of this would look like.

Consider an arbitrary resolution refutation of Peb^d_G. From this refutation we extract a pebbling of G. At some point in time t in the obtained pebbling, there must be a lot of pebbles on the vertices of G since this graph was chosen with high pebbling price. But this means that at time t, there are a lot of clauses on the blackboard. Since this holds for any resolution refutation, the refutation space of Peb^d_G must be large. The separation result now follows from the fact that pebbling contradictions are known to be refutable in linear length and constant width if d is fixed.

Unfortunately, we cannot quite get this idea to work. In the next subsection, we describe the modifications that we are forced to make and show how we can make the bits and pieces of our construction fit together to yield Theorem 1.1 and Corollary 1.2 for the special case of pyramid graphs.

2.3 Detailed overview of formal proof of space bound

The black-white pebble game played on a DAG G can be viewed as a way of proving the end result of the calculation described by G. Black pebbles denote proven partial results of the computation. White pebbles denote assumptions about partial results which have been used to derive other partial results (i. e., black pebbles), but which will have to be verified for the calculation to be complete. The final goal is a black pebble on the sink z and no other pebbles on the graph, corresponding to an unconditional proof of the end result of the calculation with any assumptions made along the way having been eliminated.

Translating this to pebbling contradictions, it turns out that a fruitful way to think of a black pebble on v is that it should correspond to truth of the disjunction $\bigvee_{i=1}^{d} x(v)_i$ of all positive literals over v, or to "truth of v." A white pebble on a vertex w can be understood to mean that we need to *assume* the partial result on w to derive some black pebble on v above w in the graph. Extending the reasoning above we get that this corresponds to an implication $\bigvee_{i=1}^{d} x(w)_i \rightarrow \bigvee_{j=1}^{d} x(v)_j$ which can be rewritten as the set of clauses

$$\left\{ \overline{x(w)}_i \vee \bigvee_{j=1}^{d} x(v)_j \;\middle|\; i \in [d] \right\} . \tag{2.1}$$

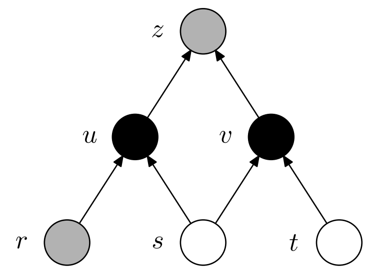
Based on this, we decide that a white-pebbled vertex should correspond to "falsity of w," i. e., to all negative literals $\overline{x(w)}_i$, $i \in [d]$, over w.

Using this intuitive correspondence, we can translate sets of clauses in a resolution refutation of Peb^d_G into black and white pebbles on G as in Figure 3. It is easy to see that if we assume $x(s)_1 \vee x(s)_2$ and $x(t)_1 \vee x(t)_2$, this assumption together with the clauses on the blackboard in Figure 3(a) imply $x(v)_1 \vee x(v)_2$, so v should be black-pebbled and s and t white-pebbled in Figure 3(b). The vertex u is also black since $x(u)_1 \vee x(u)_2$ certainly is implied by the blackboard. This translation from clauses to pebbles is quite straightforward, and seems to yield well-behaved pebblings for all "sensible" refutations of Peb^d_G.

The problem is that we have no guarantee that the resolution refutations will be "sensible." Even though it might seem more or less clear how an optimal refutation of a pebbling contradiction should proceed, a particular refutation might contain unintuitive and seemingly non-optimal derivation steps that do not make much sense from a pebble game perspective. In particular, a resolution derivation has no obvious reason always to derive truth that is restricted to single vertices. For instance, it could add the axioms $\overline{x(u)}_i \vee \overline{x(v)}_2 \vee x(z)_1 \vee x(z)_2$, $i = 1,2$, to the blackboard in Figure 3(a), derive that the truth of s and t implies the truth of either v or z, i. e., the clauses $\overline{x(s)}_i \vee \overline{x(t)}_j \vee x(v)_1 \vee x(z)_1 \vee x(z)_2$ for $i, j = 1,2$, and then erase $x(u)_1 \vee x(u)_2$ from the blackboard. Although it is hard to see from such a small example, this turns out to be a serious problem in that there appears to be no way that we can interpret

$$\begin{bmatrix} x(u)_1 \lor x(u)_2 \\ \overline{x(s)}_1 \lor \overline{x(t)}_1 \lor x(v)_1 \lor x(v)_2 \\ \overline{x(s)}_1 \lor \overline{x(t)}_2 \lor x(v)_1 \lor x(v)_2 \\ \overline{x(s)}_2 \lor \overline{x(t)}_1 \lor x(v)_1 \lor x(v)_2 \\ \overline{x(s)}_2 \lor \overline{x(t)}_2 \lor x(v)_1 \lor x(v)_2 \end{bmatrix}$$

(a) Clauses on blackboard.

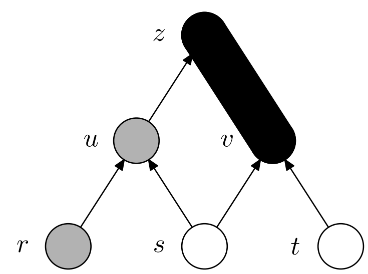


(b) Corresponding pebbles in the graph.

Figure 3: Example of intuitive correspondence between sets of clauses and pebbles.

$$\begin{bmatrix} \overline{x(s)}_1 \lor \overline{x(t)}_1 \lor x(v)_1 \lor x(z)_1 \lor x(z)_2 \\ \overline{x(s)}_1 \lor \overline{x(t)}_2 \lor x(v)_1 \lor x(z)_1 \lor x(z)_2 \\ \overline{x(s)}_2 \lor \overline{x(t)}_1 \lor x(v)_1 \lor x(z)_1 \lor x(z)_2 \\ \overline{x(s)}_2 \lor \overline{x(t)}_2 \lor x(v)_1 \lor x(z)_1 \lor x(z)_2 \end{bmatrix}$$

(a) New set of clauses on blackboard.



(b) Corresponding blobs and pebbles.

Figure 4: Interpreting sets of clauses as black blobs and white pebbles.

such derivation steps in terms of black and white pebbles without making some component in the proof idea in Section 2.2 break down.

Instead, what we do is to invent a new pebble game, with white pebbles just as before, but with black *blobs* that can cover multiple vertices instead of single-vertex black pebbles. A blob on a vertex set V can be thought of as truth of some vertex $v \in V$. The derivation sketched in the preceding paragraph, resulting in the set of clauses in Figure 4(a), will then be translated into white pebbles on s and t as before and a black blob covering both v and z in Figure 4(b). We define rules in this *blob-pebble game* corresponding roughly to black and white pebble placement and removal in the usual black-white pebble game, and add a special *inflation rule* allowing us to inflate black blobs to cover more vertices.

Once we have this blob-pebble game, we use it to construct a lower bound proof as outlined in Section 2.2. First, we establish that for a fairly general class of graphs—namely *layered* graphs, where the vertices can be divided into layers and all edges go between consecutive layers—any resolution refutation of a pebbling contradiction can be interpreted as a blob-pebbling on the DAG in terms of which this pebbling contradiction is defined. Intuitively, the reason that this works is that we can use the inflation rule to analyze apparently non-optimal steps in the refutation.

Theorem 2.1. *Let Peb^d_G denote the pebbling contradiction of degree $d \geq 1$ over a layered DAG G. Then there is a translation function from sets of clauses derived from Peb^d_G into sets of black blobs and white pebbles in G such that any resolution refutation π of Peb^d_G corresponds to a blob-pebbling $\mathcal{P}_\pi$ of G under this translation.*

In fact, the only property that we need from the layered graphs in Theorem 2.1 is that if w is a vertex with (immediate) predecessors u and v, then there is no path between the siblings u and v. The theorem holds for any DAG satisfying this condition.

Next, we carefully design a cost function for black blobs and white pebbles so that the cost of the blob-pebbling $\mathcal{P}_\pi$ in Theorem 2.1 is related to the space of the resolution refutation π.

Theorem 2.2. *If π is a refutation of a pebbling contradiction Peb^d_G of degree $d > 1$, then the cost of the associated blob-pebbling $\mathcal{P}_\pi$ is bounded by the space of π by $\mathit{cost}(\mathcal{P}_\pi) \le Sp(\pi) + \mathrm{O}(1)$.*

Without going into too much detail, in order to make the proof of Theorem 2.2 work we can only charge for black blobs having distinct lowest vertices (measured in topological order), so additional blobs with the same bottom vertices are free. Also, we can only charge for white pebbles below these bottom vertices.

Finally, we need lower bounds on blob-pebbling price. Because of the inflation rule in combination with the peculiar cost function, the blob-pebble game seems to behave rather differently from the standard black-white pebble game, and therefore we cannot appeal directly to known lower bounds on black-white pebbling price. However, for a more restricted class of graphs than in Theorem 2.1, but still including binary trees and pyramids, we manage to prove tight bounds on the blob-pebbling price by generalizing the lower bound construction for black-white pebbling in [42].

Theorem 2.3. *Any so-called* layered spreading graph *G_h of height h has blob-pebbling price $\Theta(h)$. In particular, this holds for pyramid graphs Π_h.*

Putting all of this together, we can prove our main theorem.

Theorem 1.1 (restated). *Let $Peb^d_{\Pi_h}$ denote the pebbling contradiction of degree $d > 1$ over the pyramid graph of height h. Then the clause space of refuting $Peb^d_{\Pi_h}$ by resolution is $Sp(Peb^d_{\Pi_h} \vdash \bot) = \Theta(h)$.*

Proof. The upper bound $Sp(Peb^d_{\Pi_h} \vdash \bot) = \mathrm{O}(h)$ is easy. A pyramid of height h can be pebbled with $h + \mathrm{O}(1)$ black pebbles, and a resolution refutation can mimic such a pebbling in constant extra clause space (independent of d) to refute the corresponding pebbling contradiction.

The interesting part is the lower bound. Let π be any resolution refutation of $Peb^d_{\Pi_h}$. Consider the associated blob-pebbling $\mathcal{P}_\pi$ provided by Theorem 2.1. On the one hand, we know that it holds that $\mathit{cost}(\mathcal{P}_\pi) = \mathrm{O}(Sp(\pi))$ by Theorem 2.2, provided that $d > 1$. On the other hand, Theorem 2.3 tells us that the cost of any blob-pebbling of Π_h is $\Omega(h)$, so in particular we must have $\mathit{cost}(\mathcal{P}_\pi) = \Omega(h)$. Combining these two bounds on $\mathit{cost}(\mathcal{P}_\pi)$, we see that $Sp(\pi) = \Omega(h)$. □

The pebbling contradiction Peb^d_G is a (2+d)-CNF formula and for constant d the size of the formula is linear in the number of vertices n of G (compare Figure 2). Thus, for pyramid graphs Π_h the corresponding pebbling contradictions $Peb^d_{\Pi_h}$ have size quadratic in the height h. Also, when d is fixed the upper bounds mentioned at the end of Section 2.1 become $L(Peb^d_G \vdash \bot) = \mathrm{O}(n)$ and $W(Peb^d_G \vdash \bot) = \mathrm{O}(1)$. Corollary 1.2 now follows if we set $F_n = Peb^d_{\Pi_h}$ for $d = k - 2$ and $h = \lfloor \sqrt{n} \rfloor$ and use Theorem 1.1.

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

2.4 Paper organization

Section 3 provides formal definitions of the concepts introduced in Sections 1 and 2. The bulk of the paper is then spent proving the lower-bound part of our main result in Theorem 1.1. In Section 4, we define our modified pebble game, the "blob-pebble game," that we will use to analyze resolution refutations of pebbling contradictions. In Section 5 we prove that resolution refutations can be translated into pebblings in this game, which is Theorem 2.1 in Section 2.3. In Section 6, we prove Theorem 2.2 saying that the blob-pebbling price accurately measures the clause space of the corresponding resolution refutation. Finally, after giving a fairly detailed exposition of the lower bound on black-white pebbling of [42] in Section 7 (with a somewhat simplified analysis tailor-made for our purposes), in Section 8 we delve into the details of the proof construction and modify it to apply to our blob-pebble game. This gives us Theorem 2.3. Now Theorem 1.1 and Corollary 1.2 follow as in the proofs given at the end of Section 2.3. We conclude in Section 9 by giving suggestions for further research.

3 Formal preliminaries

In this section, we define resolution, pebble games and pebbling contradictions. This is standard material and much of the discussion below is identical or very close to similar sections in [15, 47, 49].

3.1 The resolution proof system

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

A *clause* $C = a_1 \vee \cdots \vee a_k$ is a set of literals. Throughout this paper, without loss of generality all clauses C are assumed to be nontrivial in the sense that all literals in C are pairwise strictly distinct (otherwise C is trivially true since it contains some variable and its negation, and it is easy to show that such clauses can be ignored). We say that C is a *subclause* of D if $C \subseteq D$. A clause containing at most k literals is called a *k-clause*.

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

In this paper, when nothing else is stated it is assumed that A, B, C, D denote clauses; $\mathbb{C}, \mathbb{D}$ sets of clauses; x, y propositional variables; a, b, c literals; α, β truth value assignments; and ν a truth value 0 (false) or 1 (true). We write

$$\alpha^{x=\nu}(y) = \begin{cases} \alpha(y) & \text{if } y \neq x, \\ \nu & \text{if } y = x, \end{cases} \tag{3.1}$$

to denote the truth value assignment that agrees with α everywhere except possibly at x, to which it assigns the value ν. We let $\mathit{Vars}(C)$ denote the set of variables and $\mathit{Lit}(C)$ the set of literals in a clause C.[3] This notation is extended to sets of clauses by taking unions. Also, we employ the standard notation $[n] = \{1, 2, \ldots, n\}$.

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

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

$$\frac{B \vee x \quad C \vee \overline{x}}{B \vee C}\,. \tag{3.2}$$

We refer to (3.2) as *resolution on the variable* x and to $B \vee C$ as the *resolvent* of $B \vee x$ and $C \vee \overline{x}$ on x. A *resolution refutation* of a CNF formula F is a resolution derivation of the empty clause $\perp$ (the clause with no literals) from F. Perhaps somewhat confusingly, this is sometimes also referred to in the literature as a *resolution proof* of F, and we will use the two terms “proof” and “refutation” interchangeably in this paper.

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

With every resolution derivation $\pi : F \vdash A$ we can associate a DAG G_π, with the clauses in π labelling the vertices and with edges from the assumption clauses to the resolvent for each application of the resolution rule (3.2). There might be several different derivations of a clause C in π, but if so we can label each occurrence of C with a timestamp when it was derived and keep track of which copy of C is used where. A resolution derivation π is *tree-like* if any clause in the derivation is used at most once as a premise in an application of the resolution rule, i. e., if G_π is a tree. (We may make different “time-stamped” vertex copies of the axiom clauses in order to make G_π into a tree.)

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

The *width* $W(C)$ of a clause C is $|C|$, i. e., the number of literals appearing in it. The width of a set of clauses $\mathbb{C}$ is $W(\mathbb{C}) = \max_{C \in \mathbb{C}}\{W(C)\}$. The width of deriving A from F by resolution is $W(F \vdash A) = \min_{\pi : F \vdash A}\{W(\pi)\}$, and the width of refuting F is denoted $W(F \vdash \perp)$.

We next define the measure of *space*. Following the exposition in [31], a proof can be seen as a Turing machine computation, with a special read-only input tape from which the axioms can be downloaded and a working memory where all derivation steps are made. The *clause space* of a resolution proof is the maximum number of clauses that need to be kept in memory simultaneously during a verification of the proof. The *total space*[4] is the maximum total space needed, where also the width of the clauses is taken into account. For the formal definitions, it is convenient to use an alternative definition of resolution introduced in [2].

Definition 3.1 (Resolution)**.** A *clause configuration* $\mathbb{C}$ is a set of clauses. A sequence of clause configurations $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is a *resolution derivation* from a CNF formula F if $\mathbb{C}_0 = \emptyset$ and for all $t \in [\tau]$, $\mathbb{C}_t$ is obtained from $\mathbb{C}_{t-1}$ by one of the following rules:

[4]We remark that there is some terminological confusion in the literature here. In some papers, this measure has been referred to as “variable space” instead of “total space.” The terminology used here is due to Hertel and Urquhart (see [37]), and we feel that although this naming convention is as of yet less well-established, it feels much more natural.

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

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

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

A resolution derivation $\pi : F \vdash A$ of a clause A from a formula F is a derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ such that $\mathbb{C}_\tau = \{A\}$. A *resolution refutation* of F is a derivation of the empty clause $\perp$ from F.

Definition 3.2 (Clause space [2, 12])**.** The *clause space* of a resolution derivation $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is $\max_{t \in [\tau]}\{|\mathbb{C}_t|\}$. The clause space of deriving the clause A from the formula F is $Sp(F \vdash A) = \min_{\pi : F \vdash A}\{Sp(\pi)\}$, and $Sp(F \vdash \perp)$ denotes the minimum clause space of any resolution refutation of F.

Definition 3.3 (Total space [2])**.** The *total space* of a clause configuration $\mathbb{C}$ is $TotSp(\mathbb{C}) = \sum_{C \in \mathbb{C}} W(C)$. The total space of a resolution derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is $\max_{t \in [\tau]}\{TotSp(\mathbb{C}_t)\}$, and $TotSp(F \vdash \perp)$ is the minimum total space of any resolution refutation of F.

In this paper, we will be almost exclusively interested in the clause space of general, unrestricted resolution refutations. When we write simply "space" for brevity, we mean clause space in general resolution.

3.2 Pebble games and pebbling contradictions

Pebble games were originally devised for studying programming languages and compiler construction, but have later found a variety of applications in computational complexity theory. In connection with resolution, pebble games have been used both to analyze resolution derivations with respect to how much memory they consume (using the original definition of space in [31]) and to construct CNF formulas which are hard for different variants of resolution in various respects (see for example [3, 14, 19, 21]). An excellent survey of pebbling up to ca. 1980 is [51]. A second article, with more narrow focus but covering some more recent developments, is the first author's upcoming survey [48].

The *black pebbling price* of a DAG G captures the amount of memory, i. e., the number of registers, required to perform the deterministic computation described by G. The space of a non-deterministic computation is measured by the *black-white pebbling price* of G. We say that vertices of G with indegree 0 are *sources* and that vertices with outdegree 0 are *sinks* or *targets*. In the following, unless otherwise stated we will assume that all DAGs under discussion have a unique sink, and this sink will always be denoted z. The next definition is adapted from [27], though we use the established pebbling terminology introduced by [39].

Definition 3.4 (Pebble game)**.** Suppose that G is a DAG with sources S and a unique sink z. The *black-white pebble game* on G is the following one-player game. At any point in the game, there are black and white pebbles placed on some vertices of G, at most one pebble per vertex. A *pebble configuration* is a pair of subsets $\mathbb{P} = (B, W)$ of $V(G)$, comprising the black-pebbled vertices B and white-pebbled vertices W. The rules of the game are as follows:

1. If all immediate predecessors of an empty vertex v have pebbles on them, a black pebble may be placed on v. In particular, a black pebble can always be placed on any vertex in S.

2. A black pebble may be removed from any vertex at any time.

3. A white pebble may be placed on any empty vertex at any time.

4. If all immediate predecessors of a white-pebbled vertex v have pebbles on them, the white pebble on v may be removed. In particular, a white pebble can always be removed from a source vertex.

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

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

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

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

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 $[t_1,t_2]$."

The only pebblings we are really interested in are complete pebblings of G. However, when we prove lower bounds for pebbling price it will sometimes be convenient to be able to reason in terms of partial pebbling move sequences, i. e., conditional pebblings.

A *pebbling contradiction* defined on a DAG G encodes the pebble game on G by postulating the sources to be true and the sink to be false, and specifying that truth propagates through the graph according to the pebbling rules. The definition below is a generalization of formulas previously studied in [19, 53].

Definition 3.5 (Pebbling contradiction [17])**.** Suppose that G is a DAG with sources S, a unique sink z and with all non-source vertices having indegree 2, and let $d > 0$ be an integer. Associate d distinct variables $x(v)_1,\ldots,x(v)_d$ with every vertex $v \in V(G)$. The dth degree *pebbling contradiction* over G, denoted Peb^d_G, is the conjunction of the following clauses:

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

- $\overline{x(u)}_i \vee \overline{x(v)}_j \vee \bigvee_{l=1}^{d} x(w)_l$ 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*).

- $\overline{x(z)}_i$ for all $i \in [d]$ (*sink axioms* or *target axioms*),

The formula Peb^d_G is a (2+d)-CNF formula with $\mathrm{O}\big(d^2 \cdot |V(G)|\big)$ clauses over $d \cdot |V(G)|$ variables. An example pebbling contradiction is presented in Figure 2 on page 480.

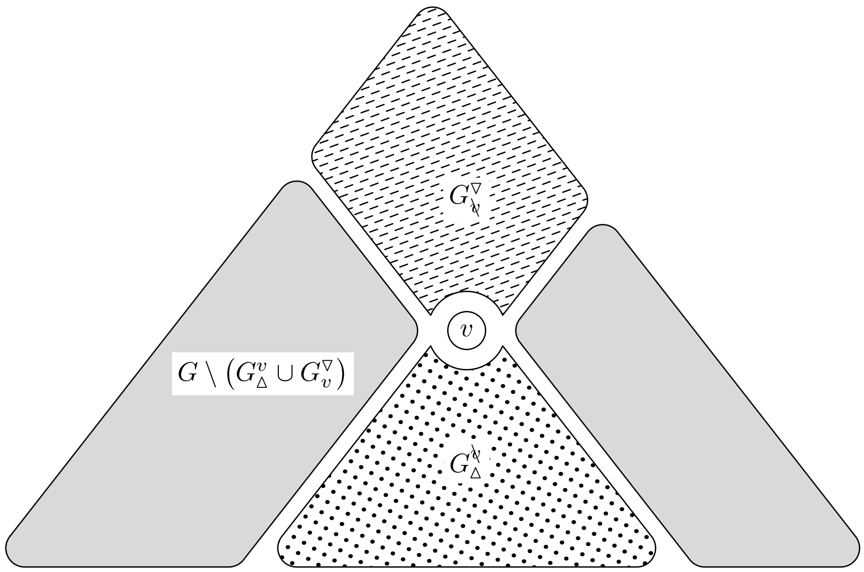


Figure 5: Notation for sets of vertices in DAG G with respect to a vertex v.

4 A game for analyzing pebbling contradictions

We now start working on the proof of Theorem 1.1, which will require the rest of this paper. In this section we present the modified pebble game that we will use to study the clause space of resolution refutations of pebbling contradictions. We remark that the game has been somewhat simplified as compared to the preliminary version [50] of this paper, incorporating some ideas from [15].

4.1 Some graph notation and definitions

We first present some notation and terminology that will be used in what follows. See Figure 5 for an illustration of the next definition.

Definition 4.1 (Graph notation)**.** We let $succ(v)$ denote the immediate successors and $pred(v)$ denote the immediate predecessors of a vertex v in a DAG G. We will usually drop the prefix "immediate" so that the terms "successor" and "predecessor" refer to an immediate successor or predecessor, respectively, unless stated otherwise.

Taking the transitive closures of $succ(\cdot)$ and $pred(\cdot)$, we let G^{∇}_{v} denote all vertices reachable from v (vertices "above" v) and $G^{v}_{\triangle}$ denote all vertices from which v is reachable (vertices "below" v). We write $G^{\not v}_{\triangle}$ and $G^{\nabla}_{\not v}$ to denote the corresponding sets with the vertex v itself removed.

If $pred(v) = \{u, w\}$, we say that u and w are *siblings*. If $u \notin G^{v}_{\triangle}$ and $v \notin G^{u}_{\triangle}$, we say that u and v are *non-comparable* vertices. Otherwise they are *comparable*.

For V a set of vertices, we let $\mathrm{bot}(V)$ denote the bottom vertices of V, i. e., the subset of vertices $\mathrm{bot}(V) = \{v \in V \mid V \cap G^{\not v}_{\triangle} = \emptyset\}$, and we let $\mathrm{top}(V) = \{v \in V \mid V \cap G^{\nabla}_{\not v} = \emptyset\}$ denote the top vertices in V.

When reasoning about arbitrary vertices we will often use as a canonical example a vertex r with assumed predecessors $pred(r) = \{p,q\}$.

Note that for a leaf v we have $pred(v) = \emptyset$, and for the sink z of G we have $succ(z) = \emptyset$. Also note that $G^v_\vartriangle$ and $G^\triangledown_v$ are sets of vertices, not subgraphs. However, we will allow ourselves to overload the notation and frequently use it for both the subgraph and its vertex set. In a similar fashion, as a rule we will overload the notation for the graph G itself and its vertices, and usually write only G when we mean $V(G)$, and when this should be clear from context.

In this paper, we will focus on *layered* directed acyclic graphs. Let us give a formal definition of this concept for completeness.

Definition 4.2 (Layered DAG)**.** A *layered DAG* is a DAG whose vertices are partitioned into (nonempty) sets of *layers* $V_0, V_1, \ldots, V_h$ on *levels* $0, 1, \ldots, h$, and whose edges run between consecutive layers. That is, if (u, v) is a directed edge, then the level of u is $L-1$ and the level of v is L for some $L \in [h]$. We say that h is the *height* of the layered DAG.

Throughout this paper, we will assume that all source vertices in a layered DAG are located on the bottom level 0. A family of layered DAGs that will be of particular interest to us are so-called *pyramid graphs*, which are defined as follows.

Definition 4.3 (Pyramid graph)**.** The *pyramid graph* Π_h of height h is a layered DAG with $h+1$ levels, where there is one vertex on the highest level (the sink z), two vertices on the next level et cetera down to $h+1$ vertices at the lowest level 0. The ith vertex at level L has incoming edges from the ith and $(i+1)$st vertices at level $L-1$.

We also need some notation for contiguous and non-contiguous topologically ordered sets of vertices in a DAG.

Definition 4.4 (Paths and chains)**.** We say that V is a *(totally) ordered* set of vertices in a DAG G, or a *chain*, if all vertices in V are comparable (i. e., if for all $u, v \in V$, either $u \in G^v_\vartriangle$ or $v \in G^u_\vartriangle$). A *path* P is a contiguous chain, i. e., such that $succ(v) \cap P \neq \emptyset$ for all $v \in P$ except the top vertex.

We write $P : v \rightsquigarrow w$ to denote a path starting in v and ending in w. A *source path* is a path that starts at some source vertex of G. A *path via* w is a path such that $w \in P$. We will also say that P *visits* w.

For a chain V we write $\mathfrak{P}_{\mathrm{via}}(V)$ denote the set of all source paths that visit all vertices in V, or that *agree* with V. Also, we write $\bigcup \mathfrak{P}_{\mathrm{via}}(V)$ for the union of all vertices in paths $P \in \mathfrak{P}_{\mathrm{via}}(V)$.

4.2 Description of the blob-pebble game and formal definition

To prove a lower bound on the refutation space of pebbling contradictions, we want to interpret derivation steps in terms of pebble placements and removals in the corresponding graph. In Section 2, we outlined an intuitive correspondence between clauses and pebbles. The problem is that if we try to use this correspondence, the pebble configurations that we get do not obey the rules of the black-white pebble game. Therefore, we are forced to to change the pebbling rules. In this section, we present the modified pebble game used for analyzing resolution derivations.

Our first modification of the pebble game is to alter the rule for white pebble removal so that a white pebble can be removed from a vertex only when a black pebble is placed on that same vertex. This will make the correspondence between pebblings and resolution derivations much more natural. Clearly, this is only a minor adjustment, and it is easy to prove formally that it does not really change anything.

Our second, and far more substantial, modification of the pebble game is motivated by the fact that in general, a resolution refutation a priori has no reason to follow our pebble game intuition. Since pebbles are induced by clauses, if at some derivation step the refutation chooses to erase "the wrong clause" from the point of view of the induced pebble configuration, this can lead to pebbles just disappearing. Whatever our translation from clauses to pebbles is, a resolution proof that suddenly out of spite erases practically all clauses must surely lead to practically all pebbles disappearing, if we want to maintain a correspondence between clause space and pebbling cost. This is all in order for black pebbles, but if we allow uncontrolled removal of white pebbles we cannot hope for any nontrivial lower bounds on pebbling price (to see this, just white-pebble the two predecessors of the sink, then black-pebble the sink itself and finally remove the white pebbles).

Our solution to this problem is 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 do the necessary bookkeeping by defining *subconfigurations* of pebbles, each subconfiguration consisting of a black pebble together with all the white pebbles this black pebble depends on, and requiring that if any pebble in a subconfiguration is removed, then all other pebbles in this subconfiguration must be removed as well.

Another problem is that resolution derivation steps can be made that appear intuitively bad given that we know that the end goal is to derive the empty clause, but where formally it appears hard to nail down wherein this supposed badness lies. To analyze such apparently non-optimal derivation steps, we introduce an *inflation* rule in which a black pebble can be inflated to a *blob* covering multiple vertices. The way to think of this is that a black pebble on a vertex v corresponds to derived truth of v, whereas for a blob pebble on V we only know that some vertex $v \in V$ is true, but not which one.

We now present the formal definition of the concept used to "label" each black blob pebble with the set of white pebbles (if any) this black pebble is dependent on. The intended meaning of the notation $[B]\langle W\rangle$ is a black blob on B together with the white pebbles W below B with the help of which we have been able to place the black blob on B. We refer to the structure $[B]\langle W\rangle$ grouping together a black blob B and its associated white pebbles W as a *blob subconfiguration*, or just *subconfiguration* for short.

Definition 4.5 (Blob subconfiguration)**.** For sets of vertices B,W in a DAG G, $[B]\langle W\rangle$ is a *blob subconfiguration* if $B \neq \emptyset$ and $B \cap W = \emptyset$. We refer to B as a (single) black *blob* and to W as (a number of different) white pebbles *supporting* B. We also say that B is *dependent* on W. If $W = \emptyset$, B is *independent*. Blobs B with $|B| = 1$ are said to be *atomic*. A set of blob subconfigurations $\mathbb{S} = \{[B_i]\langle W_i\rangle \mid i = 1,\ldots,m\}$ together constitute a *blob-pebbling configuration*.

Since the definition of the game we will play with these blobs and pebbles is somewhat involved, let us first try to give an intuitive description.

- There is one single rule corresponding to the two rules 1 and 3 for black and white pebble placement in the black-white pebble game of Definition 3.4. This *introduction* rule says that we can place a black pebble on any vertex v together with white pebbles on its predecessors (unless v is a source, in which case no white pebbles are needed).

- The analogy for rule 2 for black pebble removal in Definition 3.4 is a rule for "shrinking" black blobs. A vertex v in a blob can be eliminated by *merging* two blob subconfigurations, provided that there is both a black blob and a white pebble on v, and provided that the two black blobs involved in this *merger* do not intersect the supporting white pebbles of one another in any other vertex than v. Removing black pebbles in the black-white pebble game corresponds to shrinking atomic black blobs.

- A black blob can be *inflated* to cover more vertices, as long as it does not collide with its own supporting white vertices. Also, new supporting white pebbles can be added at an inflation move. There is no analogy of this move in the usual black-white pebble game.

- The rule 4 for white pebble removal also corresponds to merging in the blob-pebble game, in the sense that the white pebble used in the merger is eliminated as well.

- Other than that, individual white pebbles, and individual black vertices covered by blobs, can never just disappear. If we want to remove a white pebble or parts of a black blob, we can do so only by *erasing* the whole blob subconfiguration.

The formal definition follows. See Figure 6 for some examples of blob-pebbling moves.

Definition 4.6 (Blob-pebble game)**.** Let G be a DAG with unique sink z. The *blob-pebbling rules* for going from a blob-pebbling configuration $\mathbb{S}'$ to a blob-pebbling configuration $\mathbb{S}$ on G are as follows:

Introduction $\mathbb{S} = \mathbb{S}' \cup \{[v]\langle pred(v)\rangle\}$ for any $v \in V(G)$.

Merger $\mathbb{S} = \mathbb{S}' \cup \{[B]\langle W\rangle\}$ if there are $[B_1]\langle W_1\rangle, [B_2]\langle W_2\rangle \in \mathbb{S}'$ such that

1. $B_1 \cap W_2 = \emptyset$,
2. $|B_2 \cap W_1| = 1$; let v^* denote this unique element in $B_2 \cap W_1$,
3. $B = B_1 \cup (B_2 \setminus \{v^*\}) = (B_1 \cup B_2) \setminus \{v^*\}$, and
4. $W = (W_1 \setminus \{v^*\}) \cup W_2 = (W_1 \cup W_2) \setminus \{v^*\}$.

We write $[B]\langle W\rangle = \mathsf{merge}([B_1]\langle W_1\rangle, [B_2]\langle W_2\rangle)$ and refer to this as a *merger on* v^*.

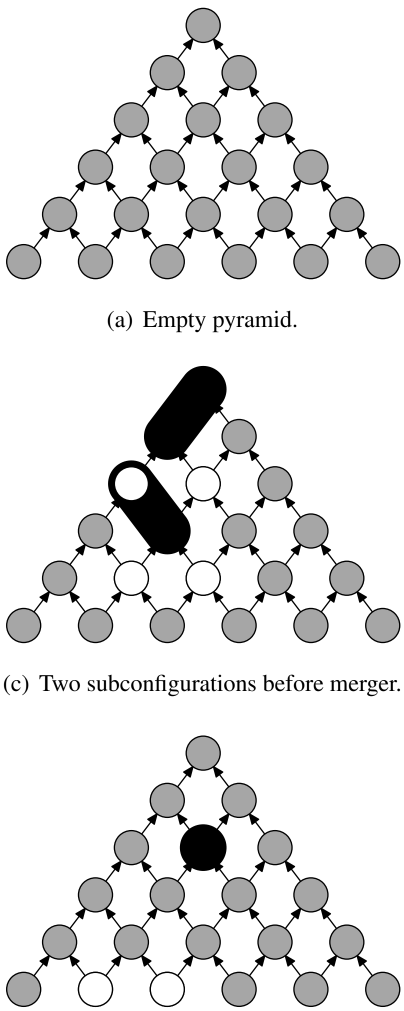
Inflation $\mathbb{S} = \mathbb{S}' \cup \{[B]\langle W\rangle\}$ if there is a $[B']\langle W'\rangle \in \mathbb{S}'$ such that $B \supseteq B'$ and $W \supseteq W'$.

We say that the blob-pebbling configuration $[B]\langle W\rangle$ is derived from $[B']\langle W'\rangle$ by *inflation* or that $[B']\langle W'\rangle$ is *inflated* to yield $[B]\langle W\rangle$.

Erasure $\mathbb{S} = \mathbb{S}' \setminus \{[B]\langle W\rangle\}$ for $[B]\langle W\rangle \in \mathbb{S}'$.

A *blob-pebbling move* at time t from $\mathbb{S}_{t-1}$ to $\mathbb{S}_t$ is either an introduction or any sequence of mergers, inflations and erasures.

For blob-pebbling configurations $\mathbb{S}_0$ and $\mathbb{S}_\tau$ on G, a *blob-pebbling* from $\mathbb{S}_0$ to $\mathbb{S}_\tau$ in G is a sequence $\mathcal{P} = \{\mathbb{S}_0, \ldots, \mathbb{S}_\tau\}$ of blob-pebbling moves. The blob-pebbling $\mathcal{P}$ is *unconditional* if $\mathbb{S}_0 = \emptyset$ and *conditional* otherwise. A *complete blob-pebbling* of G is an unconditional pebbling $\mathcal{P}$ ending in $\mathbb{S}_\tau = \{[z]\langle\emptyset\rangle\}$ for z the unique sink of G.



(a) Empty pyramid.

(b) Introduction move.

(c) Two subconfigurations before merger.

(d) The merged subconfiguration.

(e) Subconfiguration before inflation.

(f) Subconfiguration after inflation.

Figure 6: Examples of moves in the blob-pebble game.

4.3 Blob-Pebbling price

We have not yet defined what the price of a blob-pebbling is. The reason is that it is not a priori clear what the "correct" definition of blob-pebbling price should be.

It should be pointed out that the blob-pebble game has no obvious intrinsic value—its function is to serve as a tool to prove lower bounds on the resolution refutation space of pebbling contradictions. The intended structure of our lower bound proof for resolution space is that we want look at resolution refutations of pebbling contradictions, interpret them in terms of blob-pebblings on the underlying graphs, and then translate lower bounds on the price of these blob-pebblings into lower bounds on the size of the corresponding clause configurations. Therefore, we have two requirements for the blob-pebbling price $\mathsf{Blob\text{-}Peb}(G)$:

1. It should be sufficiently high, i. e., sufficiently similar to standard black-white pebbling price to enable us to prove good lower bounds on $\mathsf{Blob\text{-}Peb}(G)$, preferrably by making it possible to use lower bound proof techniques for $\mathsf{BW\text{-}Peb}(G)$ to obtain analogous bounds for $\mathsf{Blob\text{-}Peb}(G)$.

2. It should also be sufficiently low, in the sense that it should take into consideration the way subconfigurations are obtained from clauses in resolution derivations, so that lower bounds on $\mathsf{Blob\text{-}Peb}(G)$ translate back to lower bounds on the size of the clause configurations.

Hence, when defining pebbling price in Definition 4.7 below, we should also have to have in mind the coming Definition 5.2 saying how we will interpret clauses in terms of blobs and pebbles, so that these two definitions together make it possible for us to get a lower bound on clause set size in terms of pebbling cost.

For black pebbles, we could try to charge 1 for each distinct blob. But this will not work, since then the second requirement above fails. For the translation of clauses to blobs and pebbles sketched in Section 2.3 it is possible to construct clause configurations that correspond to an exponential number of distinct black blobs measured in the clause set size. The other natural extreme seems to be to charge only for mutually disjoint black blobs. But this is far too generous, and the first requirement above fails. To get a trivial example of this, take any ordinary black pebbling of G and translate in into an (atomic) blob-pebbling, but then change it so that each black pebble $[v]$ is immediately inflated to $[\{v,z\}]$ after each introduction move. It is straightforward to verify that this would yield a pebbling of G in constant cost. For white pebbles, the first idea might be to charge 1 for every white-pebbled vertex, just as in the standard pebble game. On closer inspection, though, this turns out to lead to technical problems in the proofs, and so this seems to be not quite what we need.

The definition presented below turns out to give us both of the desired properties above, and allows us to prove an optimal bound. Namely, we define blob-pebbling price so as to charge 1 for *each distinct bottom vertex* that is the unique bottom vertex of its black blob, and so as to charge for the subset of supporting white pebbles $W \cap G^b_\triangle$ in a subconfiguration $[B]\langle W\rangle$ that are *located below all bottom vertices* $\mathrm{bot}(B)$ of its black blob B. Multiple distinct blobs with the same bottom vertex come for free, however, as do blobs that do not have a unique bottom vertex. Also, any supporting white pebbles that are not "completely below" its own blob in the sense described above are also free, although we still have to keep track of them.

Definition 4.7 (Blob-pebbling price)**.** For a blob subconfiguration $[B]\langle W\rangle$, we define $\mathcal{B}([B]\langle W\rangle) = \{\mathrm{bot}(B)\}$ to be a *chargeable black vertex* if $|\mathrm{bot}(B)| = 1$ and set $\mathcal{B}([B]\langle W\rangle) = \emptyset$ otherwise. We say that $\mathcal{W}^{\triangle}([B]\langle W\rangle) = W \cap \bigcap_{b\in\mathrm{bot}(B)} G^b_{\triangle}$ are the *chargeable white vertices*. The *chargeable vertices* of the subconfiguration $[B]\langle W\rangle$ are all vertices in the union $\mathcal{B}([B]\langle W\rangle) \cup \mathcal{W}^{\triangle}([B]\langle W\rangle)$. This definition is extended to blob-pebbling configurations $\mathbb{S}$ in the natural way by letting $\mathcal{B}(\mathbb{S}) = \bigcup_{[B]\langle W\rangle\in\mathbb{S}} \mathcal{B}([B]\langle W\rangle)$ and $\mathcal{W}^{\triangle}(\mathbb{S}) = \bigcup_{[B]\langle W\rangle\in\mathbb{S}} \mathcal{W}^{\triangle}([B]\langle W\rangle)$.

The cost of a blob-pebbling configuration $\mathbb{S}$ is $\mathit{cost}(\mathbb{S}) = \left|\mathcal{B}(\mathbb{S}) \cup \mathcal{W}^{\triangle}(\mathbb{S})\right|$, and the cost of a blob-pebbling $\mathcal{P} = \{\mathbb{S}_0,\ldots,\mathbb{S}_\tau\}$ is $\mathit{cost}(\mathcal{P}) = \max_{t\in[\tau]}\{\mathit{cost}(\mathbb{S}_t)\}$.

The *blob-pebbling price* of a blob subconfiguration $[B]\langle W\rangle$, denoted $\mathit{Blob\text{-}Peb}([B]\langle W\rangle)$, is the minimal cost of any unconditional blob-pebbling $\mathcal{P} = \{\mathbb{S}_0,\ldots,\mathbb{S}_\tau\}$ such that $\mathbb{S}_\tau = \{[B]\langle W\rangle\}$. The blob-pebbling price of a DAG G is $\mathit{Blob\text{-}Peb}(G) = \mathit{Blob\text{-}Peb}([z]\langle\emptyset\rangle)$, i. e., the minimal cost of any complete blob-pebbling of G.

We will also write $\mathcal{W}(\mathbb{S})$ to denote the set of all white-pebbled vertices in $\mathbb{S}$, including non-chargeable ones.

We stress again that we make no claim that Definition 4.7 is the "obviously correct" definition of blob-pebbling price—it just happens to be a definition that works. In fact, there are other possible options, some of which are arguably more natural but lead to more complicated proofs or slightly worse bounds. To conclude this section, we just want to mention one alternative definition which seems slightly more natural to us, which yields a slightly stronger pebbling price, and which might therefore be useful if lower bounds on blob-pebbling price are to be extended from layered graphs to more general DAGs (as discussed in Section 9).

Namely, for $B_1,\ldots,B_m$ any sets of vertices, let us say that a *set of distinguished representatives* for $B_1,\ldots,B_m$ is a set $R = \{b_1,\ldots,b_m\}$ where $b_i \in B_i \setminus \bigcup_{j<i} B_j$ for all $i \in [m]$. Note that in general, such sets of distinguished representatives need not exist, but we can always find a *partial set of distinguished representatives* for $B_1,\ldots,B_m$, which we define to be a set of distinguished representatives for some (ordered) subset $B_{i_1},\ldots,B_{i_s}$ of the vertex sets. Now we can define the cost of a blob-pebbling configuration $\mathbb{S}$ to be

$$\mathit{cost}(\mathbb{S}) = \max_R \left\{\left|R \cup \mathcal{W}^{\triangle}(\mathbb{S})\right|\right\} \tag{4.1}$$

where the maximum is taken over all partial sets of distinguished representatives R for the black blobs in $\mathbb{S}$.

The proof of Theorem 6.5, which says that the clause space of a resolution refutation is lower bounded by the cost of the pebbling it induces, can be adapted to work for this definition if one proves first a lower bound for black blobs only and then a second lower bound for white pebbles only, and finally combine them in the obvious way by taking the maximum of the two bounds. Unfortunately, this loses a constant factor of 2, and for reasons explained in Section 9 we are interested in getting exactly the right multiplicative constants in this part of our argument. Therefore, in this paper we decided to stick with Definition 4.7 instead.

5 Resolution derivations induce blob-pebblings

In this section, we show how resolution refutations of pebbling contradictions can be translated to blob-pebblings (as described in Definition 4.6) of the corresponding DAGs. For simplicity, in the current section, as well as in the next one, we will write $v_1, v_2, \ldots, v_d$ instead of $x(v)_1, x(v)_2, \ldots, x(v)_d$ for the d variables associated with v in a dth degree pebbling contradiction. That is, in Sections 5 and 6 small letters with subscripts will denote variables in propositional logic only and nothing else.

It turns out that for technical reasons, it is convenient to ignore the target axioms $\bar{z}_1, \ldots, \bar{z}_d$ in a pebbling contradiction and focus on resolution derivations of $\bigvee_{l=1}^{d} z_l$ from the rest of the formula rather than resolution refutations of all of Peb_G^d. Let us write $*Peb_G^d = Peb_G^d \setminus \{\bar{z}_1, \ldots, \bar{z}_d\}$ to denote the pebbling formula over G with the target axioms in the pebbling contradiction removed. The next lemma is the formal statement saying that we may just as well study derivations of $\bigvee_{l=1}^{d} z_l$ from $*Peb_G^d$ instead of refutations of Peb_G^d.

Lemma 5.1. *For any DAG G with sink z, it holds that $Sp(Peb_G^d \vdash \bot) = Sp(*Peb_G^d \vdash \bigvee_{l=1}^{d} z_l)$.*

Proof. From any resolution derivation $\pi^* : *Peb_G^d \vdash \bigvee_{l=1}^{d} z_l$, we can obtain a resolution refutation of Peb_G^d from π^* in the same space by resolving the final clause $\bigvee_{l=1}^{d} z_l$ with all sink axioms $\bar{z}_l$, $l = 1, \ldots, d$, one by one in space 3.

In the other direction, for $\pi : Peb_G^d \vdash \bot$ 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 $\bar{z}_l$ in π, leaving the literals z_l in the clauses. Instead of the final empty clause $\bot$ we get some clause $D \subseteq \bigvee_{l=1}^{d} z_l$, and since $*Peb_G^d \nvDash D \subsetneq \bigvee_{l=1}^{d} z_l$ and resolution is sound, we have $D = \bigvee_{l=1}^{d} z_l$. ☐

In view of Lemma 5.1, from now on we will only consider resolution derivations from $*Peb_G^d$ and try to convert clause configurations in such derivations into sets of blob subconfigurations.

To avoid cluttering the notation with an excessive amount of brackets, we will sometimes use sloppy notation for sets. We will allow ourselves to omit curly brackets around singleton sets when this is clear from context, writing e. g., $V \cup v$ instead of $V \cup \{v\}$ and $[B \cup b]\langle W \cup w \rangle$ instead of $[B \cup \{b\}]\langle W \cup \{w\} \rangle$. Also, we will sometimes omit the curly brackets around sets of vertices in black blobs and write, e. g., $[u, v]$ instead of $[\{u, v\}]$.

5.1 Definition of induced configurations and theorem statement

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

$$Ax^d(r) = \left\{ \overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l \mid i, j \in [d] \right\} \tag{5.1}$$

and if r is a source, we define $Ax^d(r) = \left\{ \bigvee_{i=1}^{d} r_i \right\}$. For V a set of vertices in G, we let $Ax^d(V) = \left\{ Ax^d(v) \mid v \in V \right\}$. Note that with this notation, we have $*Peb_G^d = \left\{ Ax^d(v) \mid v \in V(G) \right\}$. For brevity, we introduce the shorthand notation

$$And^+(V) = \left\{ \bigvee_{i=1}^{d} v_i \mid v \in V \right\}, \tag{5.2a}$$

$$Or^+(V) = \bigvee_{v \in V} \bigvee_{i=1}^{d} v_i \,. \tag{5.2b}$$

The reader can think of $And^+(V)$ as "truth of all vertices in V" and $Or^+(V)$ as "truth of some vertex in V."

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 $\mathbb{C} \vDash \bot$ minimally, $\mathbb{C}$ is said to be *minimally unsatisfiable*. We say that $\mathbb{C}$ implies a clause D *maximally* if $\mathbb{C} \vDash D$ but for all $D' \subsetneqq D$ it holds that $\mathbb{C} \nvDash D'$. To define our translation of clauses to blob subconfigurations, we use implications that are in a sense both minimal and maximal.

Definition 5.2 (Induced blob subconfiguration)**.** Let G be a DAG and $\mathbb{C}$ a clause configuration derived from *Peb^d_G. Then $\mathbb{C}$ induces the blob subconfiguration $[B]\langle W\rangle$ if there is a clause set $\mathbb{D} \subseteq \mathbb{C}$ such that

$$\mathbb{D} \cup And^+(W) \vDash Or^+(B) \tag{5.3a}$$

but for which it holds for all strict subsets $\mathbb{D}' \subsetneqq \mathbb{D}$, $W' \subsetneqq W$ and $B' \subsetneqq B$ that

$$\mathbb{D}' \cup And^+(W) \nvDash Or^+(B)\,, \tag{5.3b}$$

$$\mathbb{D} \cup And^+(W') \nvDash Or^+(B)\,, \text{ and} \tag{5.3c}$$

$$\mathbb{D} \cup And^+(W) \nvDash Or^+(B')\,. \tag{5.3d}$$

We write $\mathbb{S}(\mathbb{C})$ to denote the set of all blob subconfigurations induced by $\mathbb{C}$. To save space, when all conditions (5.3a)–(5.3d) hold, we write

$$\mathbb{D} \cup And^+(W) \rhd Or^+(B) \tag{5.4}$$

and refer to this as *precise implication* or say that the clause set $\mathbb{D} \cup And^+(W)$ implies the clause $Or^+(B)$ *precisely*. Also, we say that the precise implication $\mathbb{D} \cup And^+(W) \rhd Or^+(B)$ *witnesses* the induced blob subconfiguration $[B]\langle W\rangle$.

Let us see that this definition agrees with the intuition presented in Section 2.3. An atomic black pebble on a single vertex v corresponds, as promised, to the fact that $\bigvee_{i=1}^d v_i$ is implied by the current set of clauses. A black blob on V without supporting white pebbles is induced precisely when the disjunction $Or^+(V) = \bigvee_{v\in V} \bigvee_{i=1}^d v_i$ of the corresponding clauses follow from the clauses in memory, but no disjunction over a strict subset of vertices $V' \subsetneqq V$ is implied. Finally, the supporting white pebbles just indicate that if we indeed had the information corresponding to black pebbles on these vertices, the clause corresponding to the supported black blob could be derived. Remember that our cost measure does not take into account the size of blobs. This is natural since we are interested in clause space, and since large blobs, in an intuitive sense, corresponds to large (i. e., wide) clauses rather than many clauses.

We are now ready to state the main result of this section, which says that if we apply the translation of clauses to blobs and pebbles in Definition 5.2 on all the clause configurations in a resolution derivation of $\bigvee_{l=1}^d z_l$ from *Peb^d_G, then we obtain essentially a legal blob-pebbling of G.

Theorem 5.3. *Let $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ be a resolution derivation of $\bigvee_{i=1}^d z_i$ from *Peb^d_G for a DAG G. Then the induced blob-pebbling configurations $\{\mathbb{S}(\mathbb{C}_0), \ldots, \mathbb{S}(\mathbb{C}_\tau)\}$ form the "backbone" of a complete blob-pebbling $\mathcal{P}$ of G in the sense that:*

- $\mathbb{S}(\mathbb{C}_0) = \emptyset$,
- $\mathbb{S}(\mathbb{C}_\tau) = \{[z]\langle\emptyset\rangle\}$, *and*

- *for every $t \in [\tau]$, the transition $\mathbb{S}(\mathbb{C}_{t-1}) \rightsquigarrow \mathbb{S}(\mathbb{C}_t)$ can be accomplished in accordance with the blob-pebbling rules in cost* $\max\{\mathit{cost}(\mathbb{S}(\mathbb{C}_{t-1})), \mathit{cost}(\mathbb{S}(\mathbb{C}_t))\} + \mathrm{O}(1)$.

In particular, to any derivation $\pi : {}^{*}\!Peb^d_G \vdash \bigvee_{i=1}^{d} z_i$ we can associate a complete blob-pebbling $\mathcal{P}_\pi$ of G such that $\mathit{cost}(\mathcal{P}_\pi) \leq \max_{\mathbb{C} \in \pi}\{\mathit{cost}(\mathbb{S}(\mathbb{C}))\} + \mathrm{O}(1)$.

We prove the theorem by forward induction over the derivation π. By the pebbling rules in Definition 4.6, any subconfiguration $[B]\langle W\rangle$ may be erased freely at any time. Consequently, we need not worry about subconfigurations disappearing during the transition from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$. What we do need to check, though, is that no subconfiguration $[B]\langle W\rangle$ appears inexplicably in $\mathbb{S}(\mathbb{C}_t)$ as a result of a derivation step $\mathbb{C}_{t-1} \rightsquigarrow \mathbb{C}_t$, but that we can always derive any $[B]\langle W\rangle \in \mathbb{S}(\mathbb{C}_t) \setminus \mathbb{S}(\mathbb{C}_{t-1})$ from $\mathbb{S}(\mathbb{C}_{t-1})$ by the blob-pebbling rules. Also, when several pebbling moves are needed to get from $\mathbb{S}(\mathbb{C}_t)$ to $\mathbb{S}(\mathbb{C}_{t-1})$, we need to check that these intermediate moves do not affect the pebbling cost by more than an additive constant.

The proof boils down to a case analysis of the different possibilities for the derivation step to get from $\mathbb{C}_{t-1}$ to $\mathbb{C}_t$. Since the analysis is quite lengthy, we divide it into subsections. But first of all we need some auxiliary technical lemmas.

5.2 Some technical lemmas

The next three lemmas are not hard, but will prove quite useful. We present the proofs for completeness.

Lemma 5.4. *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 $\mathbb{C}_1 = \{C \in \mathbb{C} \mid a \in Lit(C)\}$ and $\mathbb{C}_2 = \mathbb{C} \setminus \mathbb{C}_1$. Since $\mathbb{C}_2 \nvDash D$ there is a truth value assignment α such that $\alpha(\mathbb{C}_2) = 1$ and $\alpha(D) = 0$. Note that $\alpha(a) = 0$, since otherwise $\alpha(\mathbb{C}_1) = 1$ which would contradict $\mathbb{C}_1 \cup \mathbb{C}_2 = \mathbb{C} \vDash D$. It follows that $\overline{a} \notin Lit(D)$. Flip a to true and denote the resulting truth value assignment by $\alpha^{a=1}$. By construction $\alpha^{a=1}(\mathbb{C}_1) = 1$ 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 $\alpha^{a=1}(\mathbb{C}) = 1$ and $\alpha^{a=1}(D) = 0$. Contradiction. ☐

Lemma 5.5. *Suppose that C, D are clauses and $\mathbb{C}$ is a set of clauses. Then $\mathbb{C} \cup \{C\} \vDash D$ if and only if $\mathbb{C} \vDash \overline{a} \vee D$ for all $a \in Lit(C)$.*

Proof. Assume that $\mathbb{C} \cup \{C\} \vDash D$ and consider any 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 must set C to false, i. e., all $\overline{a}$ to true. Conversely, if $\mathbb{C} \vDash \overline{a} \vee D$ for all $a \in Lit(C)$ and α is such that $\alpha(\mathbb{C}) = \alpha(C) = 1$, it must hold that $\alpha(D) = 1$, since otherwise $\alpha(\overline{a} \vee D) = 0$ for some literal $a \in Lit(C)$ satisfied by α. ☐

Lemma 5.6. *Suppose that $\mathbb{C} \vDash D$ minimally. Then no literal from D can occur negated in $\mathbb{C}$, i. e., it holds that $\{\overline{a} \mid a \in Lit(D)\} \cap Lit(\mathbb{C}) = \emptyset$.*

Proof. Suppose not. Let $\mathbb{C}_1 = \{C \in \mathbb{C} \mid \exists a$ such that $\overline{a} \in Lit(C)$ and $a \in Lit(D)\}$ and let $\mathbb{C}_2 = \mathbb{C} \setminus \mathbb{C}_1$. Since $\mathbb{C}_2 \nvDash D$ there is an α such that $\alpha(\mathbb{C}_2) = 1$ and $\alpha(D) = 0$. But then $\alpha(\mathbb{C}_1) = 1$, since every $C \in \mathbb{C}_1$ contains a negated literal $\overline{a}$ from D, and these literals are all set to true by α. Contradiction. ☐

We also need the following key technical lemma connecting implication with inflation moves.

Lemma 5.7. *Let $\mathbb{C}$ be a clause set derived from $*Peb^d_G$. Suppose that $B, W \subseteq G$, $B \cap W = \emptyset$, are vertex sets such that $\mathbb{C} \cup And^+(W) \vDash Or^+(B)$. Then the blob subconfiguration $[B]\langle W\rangle$ is derivable by inflation from some $[B']\langle W'\rangle \in \mathbb{S}(\mathbb{C})$.*

Proof. Pick $\mathbb{D} \subseteq \mathbb{C}$, $W' \subseteq W$ and $B' \subseteq B$ minimal such that $\mathbb{D} \cup And^+(W') \vDash Or^+(B')$. Then $\mathbb{D} \cup And^+(W') \rhd Or^+(B')$ by definition. Note, furthermore, that $B' \neq \emptyset$ since the clause set on the left-hand side must be non-contradictory. Also, since $B' \cap W' \subseteq B \cap W = \emptyset$ we must have $\mathbb{D} \neq \emptyset$, because by Lemma 5.4 it cannot be the case that $And^+(W') \vDash Or^+(B')$. This means that $\mathbb{C}$ induces $[B']\langle W'\rangle$, and clearly $[B']\langle W'\rangle$ can be inflated to $[B]\langle W\rangle$, from which the lemma follows. □

We now start the case analysis in the proof of Theorem 5.3 for the different possible derivation steps in a resolution derivation.

5.3 Erasure

Suppose that $\mathbb{C}_t = \mathbb{C}_{t-1} \setminus \{C\}$ for $C \in \mathbb{C}_{t-1}$. It is easy to see that the only possible outcome of erasing clauses is that blob subconfigurations disappear. We note for future reference that this implies that the blob-pebbling cost decreases monotonically when going from $\mathbb{S}(\mathbb{C}_{t-1})$ to $\mathbb{S}(\mathbb{C}_t)$.

5.4 Inference

Suppose that $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{C\}$ for some clause C derived from $\mathbb{C}_{t-1}$. No blob subconfigurations can disappear at an inference step since $\mathbb{C}_{t-1} \subseteq \mathbb{C}_t$. Suppose that $[B]\langle W\rangle$ is a new subconfiguration at time t witnessed by $\mathbb{D} \cup \{C\} \cup And^+(W) \rhd Or^+(B)$ for some $\mathbb{D} \subseteq \mathbb{C}_{t-1}$. Since C is derived from $\mathbb{C}_{t-1}$, we have $\mathbb{C}_{t-1} \vDash C$. Thus, it holds that $\mathbb{C}_{t-1} \cup And^+(W) \vDash Or^+(B)$ and Lemma 5.7 tells us that $[B]\langle W\rangle$ is derivable by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$.

Since no subconfiguration disappears, the pebbling cost increases monotonically when going from $\mathbb{S}(\mathbb{C}_{t-1})$ to $\mathbb{S}(\mathbb{C}_t)$ for an inference step, which is again noted for future reference.

5.5 Axiom download

This is the interesting case. Assume that a new blob subconfiguration $[B]\langle W\rangle$ is induced at time t as the result of a download of an axiom $C \in Ax^d(r)$. Then C must be one of the clauses inducing the subconfiguration, and we get that there is a clause set $\mathbb{D} \subseteq \mathbb{C}_{t-1}$ such that

$$\mathbb{D} \cup \{C\} \cup And^+(W) \rhd Or^+(B)\,. \tag{5.5}$$

Observe that if we have $\mathbb{C}_{t-1} \cup And^+(W) \vDash Or^+(B)$ at time $t-1$, then according to Lemma 5.7 $[B]\langle W\rangle$ is derivable by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$ and we are done. Therefore, it what follows we assume the non-implication

$$\mathbb{C}_{t-1} \cup And^+(W) \nvDash Or^+(B)\,. \tag{5.6}$$

Our intuition is that download of an axiom clause $C \in Ax^d(r)$ in the resolution derivation should correspond to an introduction of $[r]\langle pred(r)\rangle$ in the induced blob-pebbling. We want to prove that any other blob subconfiguration $[B]\langle W\rangle$ in $\mathbb{S}(\mathbb{C}_t)\setminus\mathbb{S}(\mathbb{C}_t)$ can be obtained by the pebbling rules from $\mathbb{S}(\mathbb{C}_{t-1})\cup[r]\langle pred(r)\rangle$. Also, we need to prove that the pebbling moves needed to go from $\mathbb{S}(\mathbb{C}_{t-1})$ to the new configuration $\mathbb{S}(\mathbb{C}_t)$ do not increase the blob-pebbling cost by more than an additive constant compared to $\max\{\mathit{cost}(\mathbb{S}(\mathbb{C}_{t-1})),\mathit{cost}(\mathbb{S}(\mathbb{C}_t))\}=\mathit{cost}(\mathbb{S}(\mathbb{C}_t))$.

As a warm-up, let us consider the case when r is a source, i. e., $pred(r)=\emptyset$ and $C=\bigvee_{i=1}^{d} r_i$. We make a case analysis depending on whether $r\in B$ in (5.5) or not.

1. $r\in B$: It clearly holds that
$$\left\{\bigvee_{i=1}^{d} r_i\right\}\cup And^{+}(\emptyset)\rhd Or^{+}(r) \tag{5.7}$$
(the left- and right-hand sides are identical), and since the implication in (5.5) is precise, for any B such that $r\in B$ we must have $B=\{r\}$ and $W=\emptyset$. Hence $[B]\langle W\rangle=[r]\langle\emptyset\rangle$. This corresponds to an introduction for the source r at time t, which certainly is a legal blob-pebbling move.

2. $r\notin B$: In this case we can rewrite (5.5) as
$$\mathbb{D}\cup And^{+}(W\cup r)\rhd Or^{+}(B) \tag{5.8}$$
for $(W\cup r)\cap B=\emptyset$, which shows that $[B]\langle W\cup r\rangle\in\mathbb{S}(\mathbb{C}_{t-1})$. Then $[B]\langle W\rangle$ can be derived by first an introduction of $[r]\langle\emptyset\rangle$ and then a merger of this subconfiguration with $[B]\langle W\cup r\rangle$ on r.

We see that when r is a source, we can get from $\mathbb{S}(\mathbb{C}_{t-1})$ to $\mathbb{S}(\mathbb{C}_t)$ first by an introduction of $[r]\langle\emptyset\rangle$ and then possibly by some mergers on r and some erasures all combined into one pebbling move. This implies that the pebbling cost increases monotonically.

The case when r is a non-source is a bit more involved, but the general idea is the same. Suppose that $pred(r)=\{p,q\}$. Then the downloaded axiom can be written $C=\overline{p}_i\vee\overline{q}_j\vee\bigvee_{l=1}^{d} r_l$ (for some fixed indices i and j) and (5.5) becomes
$$\mathbb{D}\cup\{\overline{p}_i\vee\overline{q}_j\vee\textstyle\bigvee_{l=1}^{d} r_l\}\cup And^{+}(W)\rhd Or^{+}(B)\,, \tag{5.9}$$
where we recall that by definition, we have $B\cap W=\emptyset$. Observe that it must hold that
$$\{p,q\}\cap B=\emptyset\,, \tag{5.10}$$
since if, say, $q\in B$, we could apply Lemma 5.5 on $\overline{p}_i\vee\overline{q}_j\vee\bigvee_{l=1}^{d} r_l$ to derive that the implication $\mathbb{C}_{t-1}\cup And^{+}(W)\vDash Or^{+}(B\cup q)=Or^{+}(B)$ holds, contradicting (5.6). Also, the fact that $And^{+}(r)=\left\{\bigvee_{l=1}^{d} r_l\right\}\vDash\overline{p}_i\vee\overline{q}_j\vee\bigvee_{l=1}^{d} r_l$ implies that
$$r\notin W \tag{5.11}$$
since otherwise $And^{+}(W)\vDash\overline{p}_i\vee\overline{q}_j\vee\bigvee_{l=1}^{d} r_l=C$ and we would again have a contradiction to (5.6).

As in the case when r was a source vertex, the induction step is by a case analysis depending on whether or not $r\in B$ in the implication (5.9), which, as we recall, is (5.5) with added information about exactly what the downloaded axiom clause C looks like.

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

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

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

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

We analyze these cases in order.

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

(b) $p \in W$, $q \notin W$: In this case $[r]\langle p,q\rangle$ cannot be inflated to $[B]\langle W\rangle$, since $q \notin W$. We need to find some way to eliminate the white pebble on q.

To do this, first apply Lemma 5.5 on $\overline{q}_j \in \mathit{Lit}\big(\overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l\big)$ in (5.9) to get

$$\mathbb{D} \cup \mathit{And}^+(W) \vDash \mathit{Or}^+(B \cup q)\,. \tag{5.12}$$

Then, observing that $(B \cup q) \cap W = \emptyset$ since $q \notin W$ by assumption, use Lemma 5.7 to find $\mathbb{D}' \subseteq \mathbb{D}$, $B_q \subseteq B$, and $W_q \subseteq W$ such that

$$\mathbb{D}' \cup \mathit{And}^+(W_q) \rhd \mathit{Or}^+(B_q \cup q) \tag{5.13}$$

(where we note that the vertex q cannot disappear from the right-hand side because of the non-implication (5.6)).

In this way, we have found a subconfiguration $[B_q \cup q]\langle W_q\rangle \in \mathbb{S}(\mathbb{C}_{t-1})$ with q covered by a black blob. Using this subconfiguration, we can derive $[B]\langle W\rangle$ as follows:

- introduce $[r]\langle p,q\rangle$,
- merge $[r]\langle p,q\rangle$ and $[B_q \cup q]\langle W_q\rangle$ on q to get $[B_q \cup r]\langle W_q \cup p\rangle$,
- inflate $[B_q \cup r]\langle W_q \cup p\rangle$ to $[B_q \cup r \cup B]\langle W_q \cup p \cup W\rangle = [B]\langle W\rangle$, and
- erase $[B_q \cup r]\langle W_q \cup p\rangle$ and possibly also $[r]\langle p,q\rangle$.

In the inflation step, we used that $r \in B$ and $p \in W$ by assumption and that $B_q \subseteq B$ and $W_q \subseteq W$ by construction. Also note that the merger is correct since $r \notin W \supseteq W_q$ as noted in (5.11) and $(B_q \cup q) \cap \{p,q\} \subseteq (B \cup q) \cap \{p,q\} = q$ as noted in (5.10).

(c) $\{p,q\} \cap W = \emptyset$: Now both p and q have to be eliminated if we are to use $[r]\langle p,q\rangle$ to derive $[B]\langle W\rangle$. As in the previous case 1b, we can use Lemma 5.5 (twice) to get

$$\mathbb{D} \cup \mathit{And}^+(W) \vDash \mathit{Or}^+(B \cup p) \wedge \mathit{Or}^+(B \cup q) \tag{5.14}$$

and then appeal to Lemma 5.7 to find $[B_p \cup p]\langle W_p\rangle, [B_q \cup q]\langle W_q\rangle \in \mathbb{S}(\mathbb{C}_{t-1})$ with $B_p \cup B_q \subseteq B$ and $W_p \cup W_q \subseteq W$. Using these subconfigurations, we can perform a pebbling to get $[B]\langle W\rangle$ as follows:

- introduce $[r]\langle p,q\rangle$,
- merge $[r]\langle p,q\rangle$ with $[B_q \cup q]\langle W_q\rangle$ on q to get $[B_q \cup r]\langle W_q \cup p\rangle$,

- continue by merging $[B_q \cup r]\langle W_q \cup p\rangle$ with $[B_p \cup p]\langle W_p\rangle$ on p, resulting in the subconfiguration $[B_p \cup B_q \cup r]\langle W_p \cup W_q\rangle$,
- finally inflate $[B_p \cup B_q \cup r]\langle W_p \cup W_q\rangle$ to $[B]\langle W\rangle$ and erase the auxiliary subconfigurations used along the way.

In this final inflation, we used for the black blob that $B_p \cup B_q \subseteq B$ by construction and $r \in B$ by assumption, and for the white pebbles that $W_p \cup W_q \subseteq W$ by construction. Also, we have to check that all mergers are legal, but this follows from (5.10) and (5.11) and from how the sets B_q, W_q, B_r, W_r were constructed.

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

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

Before we do this, though, we show that there is a particular subconfiguration in $\mathbb{S}(\mathbb{C}_{t-1})$ that will be useful for us. Since $And^+(r) \vDash \overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l$, it follows from (5.9) that

$$\mathbb{D} \cup And^+(W \cup r) \vDash Or^+(B)\,, \tag{5.15}$$

and since $B \cap (W \cup r) = \emptyset$ (remember that we are now assuming $r \notin B$), we can use Lemma 5.7 to find a subconfiguration

$$[B_r]\langle W_r \cup r\rangle \in \mathbb{S}(\mathbb{C}_{t-1})\,, \quad B_r \subseteq B, W_r \subseteq W, \tag{5.16}$$

which will play an important role in the pebblings below.

(a) $\{p,q\} \subseteq W$: To derive the subconfiguration $[B]\langle W\rangle$ from $\mathbb{S}(\mathbb{C}_{t-1})$ in this case, introduce $[r]\langle p,q\rangle$ and merge with the subconfiguration $[B_r]\langle W_r \cup r\rangle \in \mathbb{S}(\mathbb{C}_{t-1})$ in (5.16) to obtain $[B_r]\langle W_r \cup \{p,q\}\rangle$. This merger is in accordance with the pebbling rules since $B_r \cap \{p,q\} \subseteq B \cap \{p,q\} = \emptyset$ according to (5.10). Then inflate $[B_r]\langle W_r \cup \{p,q\}\rangle$ to the subconfiguration $[B_r \cup B]\langle W_r \cup \{p,q\} \cup W\rangle = [B]\langle W\rangle$.

(b) $p \in W$, $q \notin W$: As in case 1b, Lemma 5.5 says that $\mathbb{D} \cup And^+(W) \vDash Or^+(B \cup q)$, and since $(B \cup q) \cap W = \emptyset$ holds by assumption we can apply Lemma 5.7 to find a subconfiguration $[B_q \cup q]\langle W_q\rangle \in \mathbb{S}(\mathbb{C}_{t-1})$ as in (5.13) such that $B_q \subseteq B$, $W_q \subseteq W$. Now do the following sequence of pebbling steps:

  - introduce $[r]\langle p,q\rangle$,
  - merge $[r]\langle p,q\rangle$ with $[B_q \cup q]\langle W_q\rangle$ on q to get $[B_q \cup r]\langle W_q \cup p\rangle$,
  - merge $[B_q \cup r]\langle W_q \cup p\rangle$ with the subconfiguration $[B_r]\langle W_r \cup r\rangle \in \mathbb{S}(\mathbb{C}_{t-1})$ in (5.16) on r to get $[B_q \cup B_r]\langle W_q \cup p \cup W_r\rangle$,
  - finally inflate $[B_q \cup B_r]\langle W_q \cup p \cup W_r\rangle$ to $[B]\langle W\rangle$ and then do the necessary erasures of auxiliary subconfigurations.

(c) $\{p,q\} \cap W = \emptyset$: As in case 1c, appeal to Lemmas 5.5 and 5.7 to find subconfigurations $[B_p \cup p]\langle W_p\rangle, [B_q \cup q]\langle W_q\rangle \in \mathbb{S}(\mathbb{C}_{t-1})$ with $B_p \cup B_q \subseteq B$ and $W_p \cup W_q \subseteq W$. Using these subconfigurations as well as $[B_r]\langle W_r \cup r\rangle \in \mathbb{S}(\mathbb{C}_{t-1})$ in (5.16), we can derive $[B]\langle W\rangle$ as follows:

- introduce $[r]\langle p,q\rangle$,
- merge $[r]\langle p,q\rangle$ with $[B_q \cup q]\langle W_q\rangle$ on the vertex q to derive $[B_q \cup r]\langle W_q \cup p\rangle$,
- merge $[B_q \cup r]\langle W_q \cup p\rangle$ with $[B_p \cup p]\langle W_p\rangle$ on the vertex p to derive the subconfiguration $[B_p \cup B_q \cup r]\langle W_p \cup W_q\rangle$,
- merge $[B_p \cup B_q \cup r]\langle W_p \cup W_q\rangle$ with $[B_r]\langle W_r \cup r\rangle$ on r resulting in the subconfiguration $[B_p \cup B_q \cup B_r]\langle W_p \cup W_q \cup W_r\rangle$,
- conclude by inflating $[B_p \cup B_q \cup B_r]\langle W_p \cup W_q \cup W_r\rangle$ to $[B]\langle W\rangle$ and erasing auxiliary subconfigurations as needed.

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

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

Summing up, to make the transition from $\mathbb{S}(\mathbb{C}_{t-1})$ to $\mathbb{S}(\mathbb{C}_t)$ in the axiom download case, we can perform the following sequence of moves (in order):

1. Introduce $[r]\langle pred(r)\rangle$.
2. Perform zero or more mergers of $[r]\langle pred(r)\rangle$ with subconfigurations in $\mathbb{S}(\mathbb{C}_{t-1})$ on some subset of the vertices $\{r\} \cup pred(r)$.
3. Make zero or more inflations of the new subconfigurations resulting from the introduction and mergers in steps 1 and 2.
4. Erase all intermediate subconfigurations derived in steps 1 and 2 and used in step 3 but not present in $\mathbb{S}(\mathbb{C}_t)$.

We need to show that the maximum cost during all of these intermediate pebbling moves is bounded in terms of $cost(\mathbb{S}(\mathbb{C}_{t-1}))$ and $cost(\mathbb{S}(\mathbb{C}_t))$. But this is immediate from the definition. Note that step 1 is one pebbling move and that the sequence of steps 2–4 counts as one move. This latter move ends in the blob-pebbling configuration $\mathbb{S}(\mathbb{C}_t)$ with $cost(\mathbb{S}(\mathbb{C}_t)) \geq cost(\mathbb{S}(\mathbb{C}_{t-1}))$. The only thing that can happen in the intermediate move is that $[r]\langle pred(r)\rangle$ is introduced and then later erased, but if so this is an extra cost of at most 3. This shows that the intermediate cost in the pebbling moves to get from $\mathbb{S}(\mathbb{C}_{t-1})$ to $\mathbb{S}(\mathbb{C}_t)$ is kept below $\max\{cost(\mathbb{S}(\mathbb{C}_{t-1})), cost(\mathbb{S}(\mathbb{C}_t))\} + 3$ which is what we need.

5.6 Wrapping up the proof of Theorem 5.3

To finish the proof, let us fix an arbitrary resolution derivation $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ of $\bigvee_{i=1}^{d} z_i$ from $*Peb_G^d$. The induction hypothesis is vacuously true at time 0, since for $\mathbb{C}_0 = \emptyset$ we clearly must have $\mathbb{S}(\mathbb{C}_0) = \emptyset$. (To argue this formally, note that we require $B \cap W = \emptyset$ and use Lemma 5.4.) At time τ, the only subconfiguration induced by $\mathbb{C}_\tau = \left\{\bigvee_{i=1}^{d} z_i\right\}$ is $\mathbb{S}(\mathbb{C}_\tau) = \{[z]\langle\emptyset\rangle\}$. Finally, we have proven in the induction step that for all $t \in [\tau]$, the blob-pebbling moves needed to accomplish the transition $\mathbb{S}(\mathbb{C}_{t-1}) \rightsquigarrow \mathbb{S}(\mathbb{C}_t)$ can be performed in cost $\max\{cost(\mathbb{S}(\mathbb{C}_{t-1})), cost(\mathbb{S}(\mathbb{C}_t))\} + 3$. Theorem 5.3 follows by induction.

6 Induced blob configurations measure clause set size

In this section, we prove that if a set of clauses $\mathbb{C}$ induces a blob-pebbling configuration $\mathbb{S}(\mathbb{C})$ according to Definition 5.2, then the cost of $\mathbb{S}(\mathbb{C})$ as specified in Definition 4.7 is at most $|\mathbb{C}|$. That is, the cost of an induced blob-pebbling configuration provides a lower bound on the size of the set of clauses inducing it. This is Theorem 6.5 below.

Observe that we cannot expect a proof of this fact to work regardless of the pebbling degree d. The induced blob-pebbling in Section 5 makes no assumptions about d, but for first-degree pebbling contradictions we know from [12] that $Sp(*Peb^1_G \vdash z_1) = Sp(Peb^1_G \vdash \bot) = \mathrm{O}(1)$. Provided $d \geq 2$, though, we show that one has to pay at least N clauses to get an induced blob-pebbling configuration of cost N.

We introduce some notation to simplify the proofs in what follows. Let us define $Vars^d(u) = \{u_1, \ldots, u_d\}$. We say that a vertex u is *represented* in a clause C derived from $*Peb^d_G$, or that C *mentions* u, if $Vars^d(u) \cap Vars(C) \neq \emptyset$. We write

$$V(C) = \left\{u \in V(G) \,\middle|\, Vars^d(u) \cap Vars(C) \neq \emptyset\right\} \tag{6.1}$$

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

$$\mathbb{C}[\![U]\!] = \{C \in \mathbb{C} \mid V(C) \cap U \neq \emptyset\} \tag{6.2}$$

to denote the subset of all clauses in $\mathbb{C}$ mentioning vertices in a vertex set U.

We now show some technical results about CNF formulas that will come in handy in the proof of Theorem 6.5. Intuitively, we will use Lemma 6.1 below together with Lemma 5.4 on page 497 to argue that if a clause set $\mathbb{C}$ induces a lot of subconfigurations, then there must be a lot of variable occurrences in $\mathbb{C}$ for variables corresponding to these vertices. Note, however, that this alone will not be enough, since this will be true also for pebbling degree $d = 1$.

Lemma 6.1. *Suppose for a set of clauses $\mathbb{C}$ and clauses D_1 and D_2 with $Vars(D_1) \cap Vars(D_2) = \emptyset$ that $\mathbb{C} \vDash D_1 \vee D_2$ but $\mathbb{C} \nvDash D_2$. Then there is a literal $a \in Lit(\mathbb{C}) \cap Lit(D_1)$.*

Proof. Pick a truth value assignment α such that $\alpha(\mathbb{C}) = 1$ but $\alpha(D_2) = 0$. Since $\mathbb{C} \vDash D$, we must have $\alpha(D_1) = 1$. Let α' be the same assignment except that all satisfied literals in D_1 are flipped to false (which is possible since they are all strictly distinct by assumption). Then $\alpha'(D_1 \vee D_2) = 0$ forces $\alpha'(\mathbb{C}) = 0$, so the flip must have falsified some previously satisfied clause in $\mathbb{C}$. □

The fact that a minimally unsatisfiable CNF formula must have more clauses than variables seems to have been proven independently a number of times (see, e. g., [1, 6, 23, 43]). We will need the following formulation of this result, relating subsets of variables in a minimally implicating CNF formula and the clauses containing variables from these subsets.

Theorem 6.2. *Suppose that F is CNF formula that implies a clause D minimally. For any subset of variables V of F, let $F_V = \{C \in F \mid Vars(C) \cap V \neq \emptyset\}$ denote the set of clauses containing variables from V. Then if $V \subseteq Vars(F) \setminus Vars(D)$, it holds that $|F_V| > |V|$. In particular, if F is a minimally unsatisfiable CNF formula, we have $|F_V| > |V|$ for all $V \subseteq Vars(F)$.*

Proof. The proof is by induction over $V \subseteq Vars(F) \setminus Vars(D)$. The base case is easy. If $|V| = 1$, then $|F_V| \geq 2$, since any $x \in V$ must occur both unnegated and negated in F by Lemma 5.4.

The inductive step just generalizes the proof of Lemma 5.4. 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 $|F_V| > |V|$, and the theorem follows. □

Continuing our intuitive argument, given that Lemmas 5.4 and 6.1 tell us that many induced subconfigurations imply the presence of many variables in $\mathbb{C}$, we will use Theorem 6.2 to demonstrate that a lot of different variable occurrences will have to translate into a lot of different clauses provided that the pebbling degree d is at least 2. Before we prove this formally, let us try to provide some intuition for why it should be true by studying two special cases. Recall the notation $And^+(V) = \left\{ \bigvee_{i \in [d]} v_i \middle| v \in V \right\}$ and $Or^+(V) = \bigvee_{v \in V} \bigvee_{i \in [d]} v_i$ from Section 5.

Example 6.3. Suppose that $\mathbb{C}$ is a clause set derived from $*Peb^d_G$ that induces N independent black blobs $B_1, \ldots, B_N$ that are pairwise disjoint, i. e., $B_i \cap B_j = \emptyset$ if $i \neq j$. Then the implications

$$\mathbb{C} \vDash Or^+(B_i) \tag{6.3}$$

hold for $i = 1, \ldots, N$. Remember that since $*Peb^d_G$ is non-contradictory, so is $\mathbb{C}$.

It is clear that a non-contradictory clause set satisfying (6.3) for all i is quite simply the set

$$\mathbb{C} = \left\{ Or^+(B_i) \middle| i = 1, \ldots N \right\} \tag{6.4}$$

consisting precisely of the clauses implied. Also, it seems plausible that this is the best one can do. Informally, if there would be strictly fewer clauses than N, some clause would have to mix variables from different blobs B_i and B_j. But then Lemma 5.4 says that there will be extra clauses needed to "neutralize" the literals from B_j in the implication $\mathbb{C} \vDash Or^+(B_i)$ and vice versa, so that the total number of clauses would have to be strictly greater than N.

As it turns out, the proof that $|\mathbb{C}| \geq N$ when $\mathbb{C}$ induces N pairwise disjoint and independent black blobs is very easy. Suppose on the contrary that (6.3) holds for $i = 1, \ldots, N$ but that $|\mathbb{C}| < N$. Let α be a satisfying assignment for $\mathbb{C}$. Choose $\alpha' \subseteq \alpha$ to be any minimal partial truth value assignment fixing $\mathbb{C}$ to true. Then for the size of the domain of α' we have $|\mathrm{Dom}(\alpha')| < N$, since at most one distinct literal is needed for every clause $C \in \mathbb{C}$ to fix it to true. This means that there is some B_i such that α' does not set any variables in $Vars^d(B_i)$. Consequently α' can be extended to an assignment α'' setting $\mathbb{C}$ to true but $Or^+(B_i)$ to false, which is a contradiction.

We remark that the above argument works for any pebbling degree including $d = 1$. Intuitively, this means that one can charge for black blobs even in the case of first degree pebbling formulas.

Example 6.4. Suppose that the clause set $\mathbb{C}$ induces an blob subconfiguration $[B]\langle W\rangle$ with $W \neq \emptyset$, and let us assume for simplicity that $\mathbb{C}$ is minimal so that the implication

$$\mathbb{C} \cup \mathit{And}^+(W) \vDash \mathit{Or}^+(B) \tag{6.5}$$

holds. We claim that $|\mathbb{C}| \geq |W| + 1$ provided that $d > 1$.

Since by definition $B \cap W = \emptyset$ we have $\mathit{Vars}(\mathit{Or}^+(B)) \cap \mathit{Vars}(\mathit{And}^+(W)) = \emptyset$, and Theorem 6.2 yields that $|\mathbb{C} \cup \mathit{And}^+(W)| \geq |\mathbb{C}[\![W]\!] \cup \mathit{And}^+(W)| > |\mathit{Vars}(\mathit{And}^+(W))|$, using the notation from (6.2). This is not quite what we want—we have a lower bound on $|\mathbb{C} \cup \mathit{And}^+(W)|$, but what we need is a bound on $|\mathbb{C}|$. But if we observe that $|\mathit{Vars}(\mathit{And}^+(W))| = d|W|$ while $|\mathit{And}^+(W)| = |W|$, we get that

$$|\mathbb{C}| \geq |\mathit{Vars}(\mathit{And}^+(W))| - |\mathit{And}^+(W)| + 1 = (d-1)|W| + 1 \geq |W| + 1 \tag{6.6}$$

as claimed.

Note that this time we had to use that $d > 1$ in order to get a lower bound on the clause set size. And indeed, it is not hard to see that a single clause of the form $C = v_1 \vee \bigvee_{w \in W} \overline{w}_1$ can induce an arbitrary number of white pebbles if $d = 1$. Intuitively, white pebbles can be had for free in degree-1 pebbling formulas.

In general, matters are more complicated than in Examples 6.3 and 6.4. If $[B_1]\langle W_1\rangle$ and $[B_2]\langle W_2\rangle$ are two induced blob subconfigurations, the black blobs B_1 and B_2 need not be disjoint, the supporting white pebbles W_1 and W_2 might also intersect, and the black blob B_1 can intersect the supporting white pebbles W_2 of the other blob. Nevertheless, if we choose with some care which vertices to charge for, the intuition provided by our examples can still be used to prove the following theorem.

Theorem 6.5. *Suppose that G is a DAG and let $\mathbb{C}$ be a set of clauses derived from the pebbling formula* $^*\mathit{Peb}^d_G$ *for $d \geq 2$. Then $|\mathbb{C}| \geq \mathit{cost}(\mathbb{S}(\mathbb{C}))$.*

Proof. Suppose that the induced set of blob subconfigurations is $\mathbb{S}(\mathbb{C}) = \big\{[B_i]\langle W_i\rangle \,\big|\, i \in [m]\big\}$. By Definition 4.7, we have $\mathit{cost}(\mathbb{S}(\mathbb{C})) = \big|\mathcal{B} \cup \mathcal{W}^{\triangle}\big|$ where

$$\mathcal{B} \;= \Big\{\mathrm{bot}(B_i) \,\Big|\, [B_i]\langle W_i\rangle \in \mathbb{S}(\mathbb{C}) \text{ and } |\mathrm{bot}(B_i)| = 1\Big\} \tag{6.7}$$

and

$$\mathcal{W}^{\triangle} = \bigcup_{[B_i]\langle W_i\rangle \in \mathbb{S}(\mathbb{C})} \left(W_i \cap \bigcap_{b \in \mathrm{bot}(B_i)} G^b_{\triangle} \right). \tag{6.8}$$

We need to prove that $|\mathbb{C}| \geq \big|\mathcal{B} \cup \mathcal{W}^{\triangle}\big|$.

We first show that all vertices in $\mathcal{B} \cup \mathcal{W}^{\triangle}$ are represented in some clause in $\mathbb{C}$. By Definition 5.2, for each $[B_i]\langle W_i\rangle \in \mathbb{S}(\mathbb{C})$ there is a clause set $\mathbb{C}_i \subseteq \mathbb{C}$ such that

$$\mathbb{C}_i \cup \mathit{And}^+(W_i) \vDash \mathit{Or}^+(B_i) \tag{6.9}$$

and such that this implication does not hold for any strict subset of $\mathbb{C}_i$, W_i or B_i. Fix (arbitrarily) such a clause set $\mathbb{C}_i$ for every $[B_i]\langle W_i\rangle \in \mathbb{S}(\mathbb{C})$ for the rest of this proof.

For the induced black blobs B_i we claim that $B_i \subseteq V(\mathbb{C}_i)$, which certainly implies $\mathcal{B} \subseteq V(\mathbb{C})$. To establish this claim, note that for any $v \in B_i$ we can apply Lemma 6.1 with $D_1 = \bigvee_{j=1}^{d} v_j$ and $D_2 = Or^+(B_i \setminus \{v\})$ on the implication (6.9), which yields that the vertex v must be represented in $\mathbb{C}_i \cup And^+(W_i)$ by some positive literal v_j. Since $B_i \cap W_i = \emptyset$, we have $Vars(And^+(W_i)) \cap Vars(Or^+(B_i)) = \emptyset$ and thus $v_j \in Lit(\mathbb{C}_i)$.

Also, we claim that $W_i \subseteq V(\mathbb{C}_i)$. To see this, note that since $B_i \cap W_i = \emptyset$ and the implication (6.9) is minimal, it follows from Lemma 5.4 that for every $w \in W_i$, all literals $\overline{w}_j$, $j \in [d]$, must be present in $\mathbb{C}_i$. Hence, in particular, it holds that $W_i \cap \bigcap_{b \in \mathrm{bot}(B_i)} G^b_{\triangle} \subseteq V(\mathbb{C}_i)$.

We now prove by induction over subsets $R \subseteq \mathcal{B} \cup \mathcal{W}^{\triangle}$ that $|\mathbb{C}[\![R]\!]| \geq |R|$. The theorem clearly follows from this since $|\mathbb{C}| \geq |\mathbb{C}[\![R]\!]|$. (The letter R is chosen to suggest that this is the set of vertices *representing* the blob-pebbling configurations $[B_i]\langle W_i\rangle \in \mathbb{S}(\mathbb{C})$ in the clause set $\mathbb{C}$.) The base case $|R| = 1$ is immediate, since we just demonstrated that all vertices $r \in R$ are represented in $\mathbb{C}$.

For the induction step, suppose that $|\mathbb{C}[\![R']\!]| \geq |R'|$ for all $R' \subsetneq R$. Pick a vertex $r \in \mathrm{top}(R)$, i. e., a "topmost" vertex $r \in R$ such that $G^{\triangledown}_{r\!\!\!\!\nwarrow} \cap R = \emptyset$. We associate a blob subconfiguration $[B_i]\langle W_i\rangle \in \mathbb{S}(\mathbb{C})$ with r as follows. If $\{r\} \in \mathrm{bot}(B_i)$ for some $[B_i]\langle W_i\rangle$, fix $[B_i]\langle W_i\rangle$ arbitrarily to such a subconfiguration. Otherwise, there must exist some $[B_i]\langle W_i\rangle$ such that $r \in W_i \cap \bigcap_{b \in \mathrm{bot}(B_i)} G^b_{\triangle}$, so fix any such subconfiguration. An important observation is that for r and $[B_i]\langle W_i\rangle$ chosen in this way it holds that

$$R \cap \bigcup_{b \in \mathrm{bot}(B_i)} G^{\triangledown}_b \subseteq \{r\}\,. \tag{6.10}$$

This is so since if $\mathrm{bot}(B_i)$ is a single vertex r, this means that all of B_i resides above this vertex in G, and if $r \in \bigcap_{b \in \mathrm{bot}(B_i)} G^b_{\triangle}$ this is just another way of saying that $\mathrm{bot}(B_i) \subseteq G^{\triangledown}_r$ and hence $B_i \subseteq G^{\triangledown}_r$.

Consider the clause set $\mathbb{C}_i \subseteq \mathbb{C}$ from (6.9) associated with $[B_i]\langle W_i\rangle$ above. Clearly, by construction $r \in V(\mathbb{C}_i)$ is one of the vertices of R mentioned by $\mathbb{C}_i$. We claim that the total number of vertices in R mentioned by $\mathbb{C}_i$ is upper-bounded by the number of clauses in $\mathbb{C}_i$ mentioning these vertices, i. e., that

$$\left|\mathbb{C}_i[\![R]\!]\right| \geq \left|R \cap V(\mathbb{C}_i)\right|. \tag{6.11}$$

Before establishing this claim, let us show that it is sufficient to prove the theorem. To this end, let us write

$$R[i] = R \cap V(\mathbb{C}_i) \tag{6.12}$$

to denote the set of all vertices in R mentioned by $\mathbb{C}_i$ and assume that $|\mathbb{C}_i[\![R]\!]| = |\mathbb{C}_i[\![R[i]]\!]| \geq |R[i]|$. Observe that $\mathbb{C}_i[\![R[i]]\!] \subseteq \mathbb{C}[\![R]\!]$, since $\mathbb{C}_i \subseteq \mathbb{C}$ and $R[i] \subseteq R$. Or in words: the set of clauses in $\mathbb{C}_i$ mentioning vertices in $R[i]$ is certainly a subset of all clauses in $\mathbb{C}$ mentioning any vertex in R. Also, by construction $\mathbb{C}_i$ does not mention any vertices in $R \setminus R[i]$ since $R[i] = R \cap V(\mathbb{C}_i)$. That is, we have

$$\mathbb{C}[\![R \setminus R[i]]\!] \subseteq \mathbb{C}[\![R]\!] \setminus \mathbb{C}_i \tag{6.13}$$

in our notation. Combining the (as of yet unproven) Claim 6.11 for $\mathbb{C}_i[\![R]\!] = \mathbb{C}_i[\![R[i]]\!]$ asserting that

$\left|\mathbb{C}_i[\![R[i]]\!]\right| \geq |R[i]|$ with the induction hypothesis for $R \setminus R[i] \subseteq R \setminus \{r\} \subsetneqq R$ we get

$$\begin{aligned}\left|\mathbb{C}[\![R]\!]\right| &= \left|\mathbb{C}_i[\![R]\!] \mathbin{\dot\cup} (\mathbb{C} \setminus \mathbb{C}_i)[\![R]\!]\right| \\ &\geq \left|\mathbb{C}_i[\![R \cap V(\mathbb{C}_i)]\!] \mathbin{\dot\cup} \mathbb{C}[\![R \setminus V(\mathbb{C}_i)]\!]\right| \\ &= \left|\mathbb{C}_i[\![R[i]]\!]\right| + \left|\mathbb{C}[\![R \setminus R[i]]\!]\right| \\ &\geq |R[i]| + |R \setminus R[i]| \\ &= |R|\end{aligned} \tag{6.14}$$

and the theorem follows by induction.

It remains to verify the Claim 6.11 that $|\mathbb{C}_i[\![R[i]]\!]| \geq |R[i]|$ for $R[i] = R \cap V(\mathbb{C}_i) \neq \emptyset$. To do so, recall first that $r \in R[i]$. Thus, $R[i] \neq \emptyset$ and if $R[i] = \{r\}$ we trivially have $|\mathbb{C}_i[\![R[i]]\!]| \geq 1 = |R[i]|$. Suppose therefore that $R[i] \supsetneqq \{r\}$. We want to apply Theorem 6.2 on the formula $F = \mathbb{C}_i \cup \mathit{And}^+(W_i)$ on the left-hand side of the minimal implication (6.9). Let $R' = R[i] \setminus \{r\}$, write $R' = R_1 \mathbin{\dot\cup} R_2$ for $R_1 = R' \cap W_i$ and $R_2 = R' \setminus R_1$, and consider the subformula

$$\begin{aligned}F_{R'} &= \left\{C \in \left(\mathbb{C}_i \cup \mathit{And}^+(W_i)\right) \middle| V(C) \cap R' \neq \emptyset\right\} \\ &= \mathbb{C}_i[\![R']\!] \cup \mathit{And}^+(R_1)\end{aligned} \tag{6.15}$$

of $F = \mathbb{C}_i \cup \mathit{And}^+(W_i)$. A key observation for the concluding part of the argument is that by (6.10) we have $\mathit{Vars}^d(R') \cap \mathit{Vars}(\mathit{Or}^+(B_i)) = \emptyset$.

For each $w \in R_1$, the clauses in $\mathit{And}^+(R_1)$ contain d literals $w_1, \ldots, w_d$ and these literals must all occur negated in $\mathbb{C}_i$ by Lemma 5.4. For each $u \in R_2$, the clauses in $\mathbb{C}_i[\![R']\!]$ contain at least one variable u_i. Appealing to Theorem 6.2 with clause $D = \mathit{Or}^+(B_i)$ and variable subset $V = \mathit{Vars}^d(R') \cap \mathit{Vars}(\mathbb{C}_i) \subseteq \mathit{Vars}(F) \setminus \mathit{Vars}(\mathit{Or}^+(B_i))$, we get

$$\begin{aligned}\left|F_{R'}\right| &= \left|\mathbb{C}_i[\![R']\!] \cup \mathit{And}^+(R_1)\right| \\ &\geq \left|\mathit{Vars}^d(R') \cap \mathit{Vars}(\mathbb{C}_i)\right| + 1 \\ &\geq d\left|R_1\right| + \left|R_2\right| + 1\,,\end{aligned} \tag{6.16}$$

and rewriting this as

$$\begin{aligned}\left|\mathbb{C}_i[\![R[i]]\!]\right| &\geq \left|\mathbb{C}_i[\![R']\!]\right| \\ &= \left|F_{R'}\right| - \left|\mathit{And}^+(R_1)\right| \\ &\geq (d-1)\left|R_1\right| + \left|R_2\right| + 1 \\ &\geq \left|R[i]\right|\end{aligned} \tag{6.17}$$

establishes the claim. □

We have two concluding remarks. Firstly, we note that the place where the condition $d \geq 2$ is needed is the very final step (6.17). This is where an attempted lower bound proof for first degree pebbling formulas $*\mathit{Peb}^1_G$ would fail for the reason that the presence of many white pebbles in $\mathbb{S}(\mathbb{C})$ says absolutely nothing about the size of the clause set $\mathbb{C}$ inducing these pebbles. Secondly, another crucial step in the proof is that we can choose our representative vertices $r \in R$ so that (6.10) holds. It is thanks to this that the inequalities in (6.16) go through. The way we make sure (6.10) holds is to charge only for (distinct) bottom vertices in the black blobs, and only for supporting white pebbles below these bottom vertices.

7 Black-white pebbling and layered graphs

Having come this far in the paper, we know that resolution derivations induce blob-pebblings. We also know that blob-pebbling cost gives a lower bound on clause set size and hence on the space of the derivation. The final component needed to make the proof of Theorem 1.1 complete is to show lower bounds on the blob-pebbling price $\mathit{Blob\text{-}Peb}(G_i)$ for some nice family of DAGs $\{G_i\}_{i=1}^{\infty}$.

Perhaps the first idea that comes to mind is to try to establish lower bounds on blob-pebbling price by reducing this problem to the problem of proving lower bounds for the standard black-white pebble game of Definition 3.4. This is what is done in [47] for the restricted case of trees. There, for the pebblings $\mathcal{P}_\pi$ that one gets from resolution derivations $\pi : {}^*Peb_T^d \vdash \bigvee_{i=1}^{d} z_i$ in a rather different so-called "labelled" pebble game, an explicit procedure is presented to transform $\mathcal{P}_\pi$ into a complete black-white pebblings of T in essentially the same cost except for a small constant-factor blow-up. The lower bound on pebbling price in the labelled pebble game then follows immediately by using the known lower bound for black-white pebbling of trees in [44].

Unfortunately, the blob-pebble game seems more difficult than the game in [47] to analyze in terms of the standard black-white pebble game. The problem is the inflation rule in combination with the cost function. It is straightforward to show that without inflation, the blob-pebble game is essentially just a disguised form of black-white pebbling. Thus, if we could convert any blob-pebbling into an equivalent pebbling not using inflation moves without increasing the cost by more than, say, some constant factor, we would be done. But in contrast to the case for the labelled pebble game in [47] played on binary trees, it seems hard to transform blob-pebblings into black-white pebblings in a cost-preserving way.

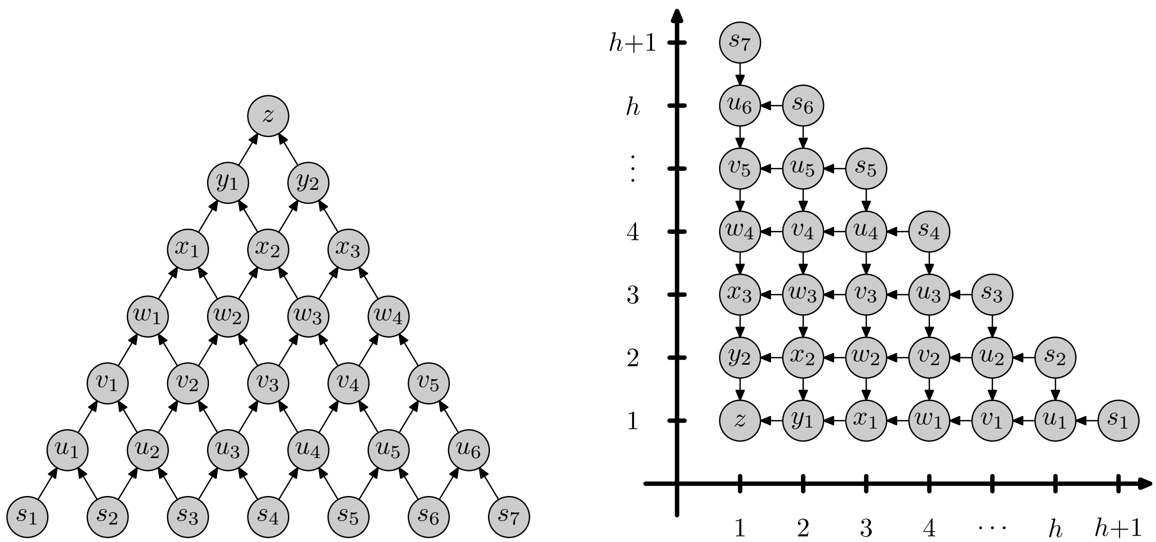
Instead, what we do is to prove lower bounds directly for the blob-pebble game. This is not immediately clear how to do, since the lower bound proofs for black-white pebbling price in, e. g., [27, 35, 42, 44] all break down for the more general blob-pebble game. And indeed, to obtain lower bounds we have to limit our attention to the class of *layered spreading graphs* (to be defined below), a class that includes binary trees and pyramid graphs.

In our proof, we borrow heavily from the corresponding bound for black-white pebbling by Klawe [42], but we need to go quite deep into the construction in order to make the changes necessary for the proof go through in the blob-pebbling case. In this section, we therefore give a detailed exposition of the lower bound in [42]. We believe that an in-depth description of this construction, which in addition has been somewhat simplified as compared to the original paper, is crucial in order to give the reader a fair chance to understand what is going on in the next section, where we generalize the bound from the black-white pebble game to the blob-pebble game in Definition 4.6.

7.1 Some preliminaries and a tight bound for black pebbling

Unless otherwise stated, in the following G denotes a layered DAG (Definition 4.2); u,v,w,x,y,z denote vertices of G, where as before z is assumed to be the unique sink vertex; U,V,W,X,Y denote sets of vertices; P denotes a path; and $\mathfrak{P}$ denotes a set of paths. We will also use the following notation.

Definition 7.1 (Layered DAG notation)**.** For a vertex u in a layered DAG G we let $\mathrm{level}(u)$ denote the level of u. For a vertex set U we let $\mathrm{minlevel}(U) = \min\{\mathrm{level}(u) : u \in U\}$ and $\mathrm{maxlevel}(U) =$



(a) Pyramid graph of height $h = 6$.

(b) Pyramid as fragment of 2D rectilinear lattice.

Figure 7: The pyramid Π_6 of height 6 with labelled vertices.

$\max\{\text{level}(u) : u \in U\}$ denote the lowest and highest level, respectively, of any vertex in U. Vertices in U on particular levels are denoted as follows:

- $U\{\succeq j\} = \{u \in U \mid \text{level}(u) \geq j\}$ denotes the subset of all vertices in U on level j or higher.
- $U\{\succ j\} = \{u \in U \mid \text{level}(u) > j\}$ denotes the vertices in U strictly above level j.
- $U\{\sim j\} = U\{\succeq j\} \setminus U\{\succ j\}$ denotes the vertices exactly on level j.

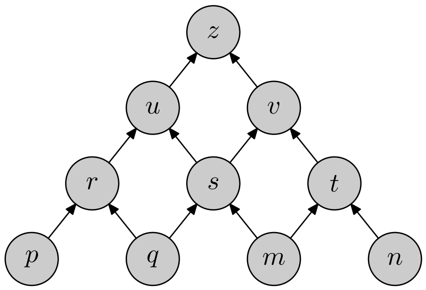
The vertex sets $U\{\preceq j\}$ and $U\{\prec j\}$ are defined completely analogously.

For the layered DAGs G under consideration we will assume that all sources are on level 0 and that all non-sources have indegree 2.

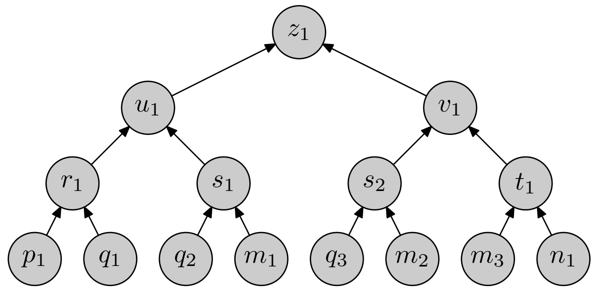
Although most of what will be said in what follows holds for arbitrary layered DAGs, we will focus on pyramids since these are the graphs that we are most interested in. Figure 7(a) presents a pyramid graph with labelled vertices that we will use as a running example. Pyramid graphs can also be visualized as triangular fragments of a directed two-dimensional rectilinear lattice. Perhaps this can sometimes make it easier for the reader to see that "obvious" statements about properties of pyramids in some of the proofs below are indeed obvious. In Figure 7(b), the pyramid in Figure 7(a) is redrawn as such a lattice fragment.

In the standard black and black-white pebble games, we have the following upper bounds on pebbling price of layered DAGs.

Lemma 7.2. *For any layered DAG G_h of height h with a unique sink z and all non-sources having vertex indegree 2, it holds that $\mathit{Peb}(G_h) \leq h + \mathrm{O}(1)$ and $\mathit{BW}\text{-}\mathit{Peb}(G_h) \leq h/2 + \mathrm{O}(1)$.*



(a) Pyramid graph Π_3 of height 3.



(b) Binary tree T_3 with vertex labels from Π_3.

Figure 8: Binary tree with vertices labelled by pyramid graph vertices as in proof of Lemma 7.2.

Proof. The bounds above are true for complete binary trees of height h according to [44]. It is not hard to see that the corresponding pebbling strategies can be used to pebble any layered graph of the same height with at most the same amount of pebbles.

Formally, suppose that the sink z of the DAG G_h has predecessors x and y. Label the root of T_h by z_1 and its predecessors by x_1 and y_1. Recursively, for a vertex in T_h labelled by w_i, look at the corresponding vertex w in G_h and suppose that $pred(w) = \{u, v\}$. Then label the vertices $pred(w_i)$ in T_h by u_j and v_k for the smallest positive indices j, k such that there are not already other vertices in T_h labelled u_j and v_k. In Figure 8 there is an illustration of how the vertices in a pyramid Π_3 of height 3 are mapped to vertices in the complete binary tree T_3 in this manner.

The result is a labelling of T_h where every vertex v in G_h corresponds to one or more distinct vertices $v_1, \ldots, v_{k_v}$ in T_h, and such that if $pred(w_i) = \{u_j, v_k\}$ in T_h, then $pred(w) = \{u, v\}$ in G_h. Given a pebbling strategy $\mathcal{P}$ for T_h, we can pebble G_h with at most the same amount of pebbles by mimicking any move on any v_i in T_h by performing the same move on v in G_h. The details are easily verified. □

In this section, we will identify some layered graphs G_h for which the bound in Lemma 7.2 is also the asymptotically correct lower bound. As a warm-up, and also to introduce some important ideas, let us consider the black pebbling price of the pyramid Π_h of height h.

Theorem 7.3 ([25])**.** *$\mathsf{Peb}(\Pi_h) = h + 2$ for $h \geq 1$.*

To prove this lower bound, it turns out that it is sufficient to study blocked paths in the pyramid.

Definition 7.4. A vertex set U *blocks* a path P if $U \cap P \neq \emptyset$. A vertex set U blocks a set of paths $\mathfrak{P}$ if U blocks all $P \in \mathfrak{P}$.

Proof of Theorem 7.3. It is easy to devise (inductively) a black pebbling strategy that uses $h+2$ pebbles (using, e. g., Lemma 7.2). We show that this is also a lower bound.

Consider the first time t when all possible paths from sources to the sink are blocked by black pebbles (there is such a time since the black pebble on the sink blocks all paths at the end of the pebbling). Suppose that P is one previously open path that was blocked at time t. Then P must be blocked by placing a pebble on some source vertex u. The path P contains $h+1$ vertices, and for each vertex $v \in P \setminus \{u\}$ there is a unique path P_v that coincides with P from v onwards to the sink but arrives at v in a straight

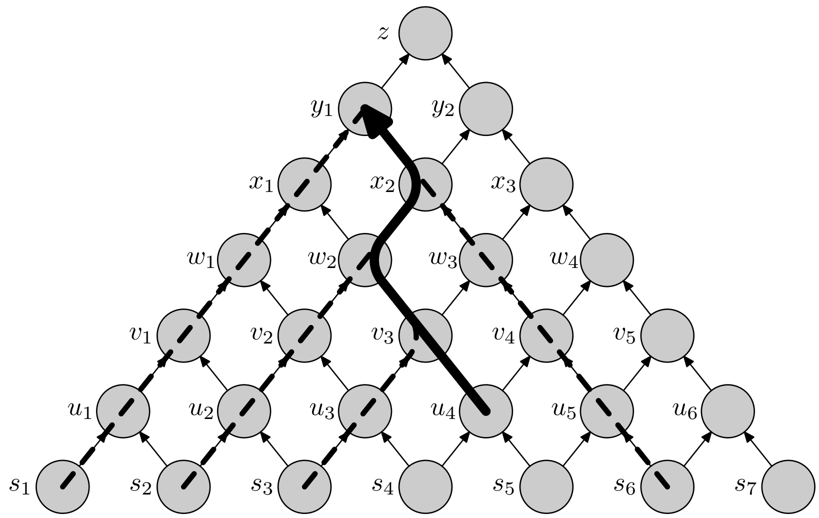


Figure 9: Set of converging source paths (dashed) for the path $P : u_4 \rightsquigarrow y_1$ (solid).

line from a source "in the opposite direction" of that of P, i. e., via the immediate predecessor of v not contained in P. At time $t-1$ all such paths $\{P_v \mid v \in P \setminus \{u\}\}$ must already be blocked, and since P is still open no pebble can block two paths $P_v \neq P_{v'}$ for $v, v' \in P \setminus \{u\}$, $v \neq v'$. Thus at time t there are at least $h+1$ pebbles on Π_h. Furthermore, without loss of generality each pebble placement on a source vertex is followed by another pebble placement (otherwise perform all removals immediately following after time t before making the pebble placement at time t). Thus at time $t+1$ there are $h+2$ pebbles on Π_h. □

We will use the idea in the proof above about a set of paths converging at different levels to another fixed path repeatedly, so we write it down as a separate observation.

Observation 7.5. *Suppose that u and w are vertices in Π_h on levels $L_u < L_w$ and that $P : u \rightsquigarrow w$ is a path from u to w. Let $K = L_w - L_v$ and write $P = \{v_0 = u, v_1, \dots, v_K = w\}$. Then there is a set of K paths $\mathfrak{P} = \{P_1, \dots, P_K\}$ such that P_i coincides with P from v_i onwards to w and arrives to v_i in a straight line from a source vertex via the immediate predecessor of v_i which is not contained in P, i. e., is distinct from v_{i-1}. In particular, for any i, j with $1 \leq i < j \leq k$ it holds that $P_i \cap P_j \subseteq P_j \cap P \subseteq P \setminus \{u\}$.*

We will refer to the paths $P_1, \dots, P_K$ as a set of *converging source paths*, or just converging paths, for $P : u \rightsquigarrow w$. See Figure 9 for an example.

7.2 A tight bound on the black-white pebbling price of pyramids

The rest of this section contains an exposition of the black-white pebbling price lower bound for pyramids by Klawe [42], which we will adapt in Section 8 to work for the blob-pebble game. We have slightly modified (and hopefully simplified) the proofs, and have also had to change much of the notation and terminology in [42] to agree better with this the rest of this paper in general and the blob-pebble game in particular. Perhaps the most important change is that we restrict all definitions to layered graphs, in contrast to Klawe who deals with a somewhat more general class of DAGs. We concentrate on layered graphs mainly to avoid unnecessary complications in the exposition, and since it can be proven that no graphs in [42] can give a better size/pebbling price trade-off than one gets for layered graphs anyway.

Recall from Definition 4.4 that a *path via w* is a path P such that $w \in P$. We will also say that P *visits* w. The notation $\mathfrak{P}_{\mathrm{via}}(w)$ is used to denote all source paths visiting w. Note that a path $P \in \mathfrak{P}_{\mathrm{via}}(w)$ visiting w may continue after w, or may end in w.

Definition 7.6 (Hiding set)**.** A vertex set U *hides* a vertex w if U blocks all source paths visiting w, i. e., if U blocks $\mathfrak{P}_{\mathrm{via}}(w)$. U hides W if U hides all $w \in W$. If so, we say that U is a *hiding set* for W. We write $\llbracket U \rrbracket$ to denote the set of all vertices hidden by U.

Our perspective is that we are standing at the sources of G and looking towards the sink. Then U *hides* w if we "cannot see" w from the sources since U covers all paths approaching w (in particular, a vertex w hides itself). When U *blocks* a path P is is possible that we can "see" the beginning of the path, but we cannot walk all of the path since it is blocked somewhere on the way. The reason why this terminological distinction is convenient will hopefully become clear in the next section.

Note that if U should hide w, then in particular it must block all paths ending in w. Therefore, when looking at minimal hiding sets we can assume without loss of generality that no vertex in U is on a level higher than w.

It is an easy exercise to show that the hiding relation is transitive, i. e., that if U hides V and V hides W, then U hides W. Let us state this as a proposition for reference.

Proposition 7.7. *If* $V \subseteq \llbracket U \rrbracket$ *and* $W \subseteq \llbracket V \rrbracket$ *then* $W \subseteq \llbracket U \rrbracket$.

One key concept in Klawe's paper is that of *potential*. The potential of a black-white pebble configuration $\mathbb{P} = (B, W)$ is intended to measure how "good" the configuration $\mathbb{P}$ is, or at least how hard it is to reach in a pebbling. Note that this is not captured by the cost of the current pebble configuration. For instance, the final configuration $\mathbb{P}_\tau = (\{z\}, \emptyset)$ is the best configuration conceivable, but only costs 1. At the other extreme, the configuration $\mathbb{P}$ in a pyramid with, say, all vertices on level L white-pebbled and all vertices on level $L+1$ black-pebbled is potentially very expensive (for low levels L), but does not seem very useful. Since this configuration on the one hand is quite expensive, but on the other hand is extremely easy to derive (just white-pebble all vertices on level L, and then black-pebble all vertices on level $L+1$), here the cost seems like a gross overestimation of the "goodness" of $\mathbb{P}$.

Klawe's potential measure remedies this. The potential of a pebble configuration (B, W) is declared to be the minimum *measure*, as defined next, of any set U that together with W hides B. Recall that $U\{\succeq j\}$ denotes the subset of all vertices in U on level j or higher in a layered graph G.

Definition 7.8 (Measure)**.** The *jth partial measure* of the vertex set U in G is

$$m_G^j(U) = \begin{cases} j + 2|U\{\succeq j\}| & \text{if } U\{\succeq j\} \neq \emptyset, \\ 0 & \text{otherwise,} \end{cases}$$

and the *measure* of U is $m_G(U) = \max_j \{ m_G^j(U) \}$.

Definition 7.9 (Potential)**.** We say that U is a hiding set for a black-white pebble configuration $\mathbb{P} = (B,W)$ in a layered graph G if $U \cup W$ hides B. We define the *potential* of the pebble configuration to be

$$\mathrm{pot}_G(\mathbb{P}) = \mathrm{pot}_G(B,W) = \min\{m_G(U) : U \text{ is a hiding set for } (B,W)\}\,.$$

If U is a hiding set for (B,W) with minimal measure $m_G(U)$ among all vertex sets U' such that $U' \cup W$ hides B, we say that U is a *minimum-measure* hiding set for $\mathbb{P}$.

Since the graph under consideration will almost always be clear from context, we will tend to omit the subindex G in measures and potentials.

We remark that although this might not be immediately obvious, there is quite a lot of nice intuition why Definition 7.9 should be a relevant estimate of how "good" a pebble configuration is. We refer the reader to Section 2 of [42] for a discussion about this. Let us just note that with this definition, the pebble configuration $\mathbb{P}_\tau = (\{z\},\emptyset)$ has high potential, as we shall soon see, while the configuration with all vertices on level L white-pebbled and all vertices on level $L+1$ black-pebbled has potential zero. This agrees with our intuitive understanding as explained before Definition 7.8.

Remark 7.10. Klawe does not use the level of a vertex u in Definitions 7.8 and 7.9, but instead the black pebbling price $\mathit{Peb}(\{u\},\emptyset)$ of the configuration with a black pebble on u and no other pebbles in the DAG. For pyramids, these two concepts are equivalent, and we feel that the exposition can be made considerably simpler by using levels.

Klawe proves two facts about the potentials of the pebble configurations in any black-white pebbling $\mathcal{P} = \{\mathbb{P}_0,\dots,\mathbb{P}_\tau\}$ of a pyramid graph Π_h:

1. The potential correctly estimates the goodness of the current configuration $\mathbb{P}_t$ by taking into account the whole pebbling that has led to $\mathbb{P}_t$. Namely, $\mathrm{pot}(\mathbb{P}_t) \le 2 \cdot \max_{s \le t}\{\mathit{cost}(\mathbb{P}_s)\}$.
2. The final configuration $\mathbb{P}_\tau = (\{z\},\emptyset)$ has high potential, namely $\mathrm{pot}(\{z\},\emptyset) = h + \mathrm{O}(1)$.

Combining these two parts, one clearly gets a lower bound on pebbling price.

For pyramids, part 2 is not too hard to show directly. In fact, it is a useful exercise if one wants to get some feeling for how the potential works. Part 1 is much trickier. It is proven by induction over the pebbling. As it turns out, the whole induction proof hinges on the following key property.

Property 7.11 (Limited hiding-cardinality property)**.** We say that the black-white pebble configuration $\mathbb{P} = (B,W)$ in G has the *limited hiding-cardinality property*, or just the *LHC property* for short, if there is a vertex set U such that

1. U is a hiding set for $\mathbb{P}$,
2. $\mathrm{pot}_G(\mathbb{P}) = m(U)$,
3. $U = B$ or $|U| < |B| + |W| = \mathit{cost}(\mathbb{P})$.

We say that the graph G has the limited hiding-cardinality property if all black-white pebble configurations $\mathbb{P} = (B,W)$ on G have the limited hiding-cardinality property.

Note that requirements 1 and 2 just say that U is a vertex set that witnesses the potential of $\mathbb{P}$. The important point here is requirement 3, which says (basically) that if we are given a hiding set U with minimum measure but with size exceeding the cost of the black-white pebble configuration $\mathbb{P}$, then we can pick *another* hiding set U' which keeps the minimum measure but decreases the cardinality to at most $\mathit{cost}(\mathbb{P})$.

Given Property 7.11, the induction proof for part 1 follows quite easily. The main part of the paper [42] is then spent on proving that a class of DAGs including pyramids have Property 7.11. Let us see what the lower bound proof looks like, assuming that Property 7.11 holds.

Lemma 7.12 (Theorem 2.2 in [42]). *Let G be a layered graph possessing the LHC property and suppose that $\mathcal{P} = \{\mathbb{P}_0 = \emptyset, \mathbb{P}_1, \ldots, \mathbb{P}_\tau\}$ is any unconditional black-white pebbling on G. Then it holds for all $t = 1, \ldots, \tau$ that $\mathrm{pot}_G(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathit{cost}(\mathbb{P}_s)\}$.*

Proof. To simplify the proof, let us assume without loss of generality that no white pebble is ever removed from a source. If $\mathcal{P}$ contains such moves, we just substitute for each such white pebble placement on v a black pebble placement on v instead, and when the white pebble is removed we remove the corresponding black pebble. It is easy to check that this results in a legal pebbling $\mathcal{P}'$ that has exactly the same cost.

The proof is by induction. The base case $\mathbb{P}_0 = \emptyset$ is trivial. For the induction hypothesis, suppose that $\mathrm{pot}(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathit{cost}(\mathbb{P}_s)\}$ and let U_t be a vertex set as in Property 7.11, i. e., such that $U_t \cup W_t$ hides B_t, $\mathrm{pot}(\mathbb{P}_t) = m(U_t)$ and $|U_t| \leq \mathit{cost}(\mathbb{P}_t) = |B| + |W|$.

Consider $\mathbb{P}_{t+1}$. We need to show that $\mathrm{pot}(\mathbb{P}_{t+1}) \leq 2 \cdot \max_{s \leq t+1}\{\mathit{cost}(\mathbb{P}_s)\}$. By the induction hypothesis, it is sufficient to show that

$$\mathrm{pot}(\mathbb{P}_{t+1}) \leq \max\{\mathrm{pot}(\mathbb{P}_t), 2 \cdot \mathit{cost}(\mathbb{P}_{t+1})\}\,. \tag{7.1}$$

We also note that if $U_t \cup W_{t+1}$ hides B_{t+1} we are done, since if so $\mathrm{pot}(\mathbb{P}_{t+1}) \leq m(U_t) = \mathrm{pot}(\mathbb{P}_t)$. We make a case analysis depending on the type of move made to get from $\mathbb{P}_t$ to $\mathbb{P}_{t+1}$.

1. Removal of black pebble: In this case, $U_t \cup W_{t+1} = U_t \cup W_t$ obviously hides $B_{t+1} \subset B_t$ as well, so $\mathrm{pot}(\mathbb{P}_{t+1}) \leq \mathrm{pot}(\mathbb{P}_t)$.

2. Placement of white pebble: Again, $U_t \cup W_{t+1} \supset U_t \cup W_t$ hides $B_{t+1} = B_t$, so $\mathrm{pot}(\mathbb{P}_{t+1}) \leq \mathrm{pot}(\mathbb{P}_t)$.

3. Removal of white pebble: Suppose that a white pebble is removed from the vertex w, so $W_{t+1} = W_t \setminus \{w\}$. As noted above, without loss of generality w is not a source vertex. We claim that $U_t \cup W_{t+1}$ still hides $B_{t+1} = B_t$, from which $\mathrm{pot}(\mathbb{P}_{t+1}) \leq \mathrm{pot}(\mathbb{P}_t)$ follows as above.

   To see that the claim is true, note that $\mathit{pred}(w) \subseteq B_t \cup W_t$ by the pebbling rules, for otherwise we would not be able to remove the white pebble on w. If $\mathit{pred}(w) \subseteq W_t$ we are done, since then $U_t \cup W_{t+1}$ hides $U_t \cup W_t$ and we can use the transitivity in Proposition 7.7. If instead there is some $v \in \mathit{pred}(w) \cap B_t$, then $U_t \cup W_t = U_t \cup W_{t+1} \cup \{w\}$ hides v by assumption. Since w is a successor of v, and therefore on a higher level than v, we must have $U_t \cup W_t \setminus \{w\}$ hiding v. Thus in any case $U_t \cup W_{t+1}$ hides $\mathit{pred}(w)$, so by transitivity $U_t \cup W_{t+1}$ hides B_{t+1}.

4. Placement of black pebble: Suppose that a black pebble is placed on v. If v is not a source, by the pebbling rules we again have that $\mathit{pred}(v) \subseteq B_t \cup W_t$. In particular, $B_t \cup W_t$ hides v and by transitivity we have that $U_t \cup W_{t+1} = U_t \cup W_t$ hides $B_t \cup \{v\} = B_{t+1}$.

The case when v is a source turns out to be the only interesting one. Now $U_t \cup W_t$ does not necessarily hide $B_t \cup \{v\} = B_{t+1}$ any longer. An obvious fix is to try with $U_t \cup \{v\} \cup W_t$ instead. This set clearly hides B_{t+1}, but it can be the case that $m(U_t \cup \{v\}) > m(U_t)$. This is problematic, since we could have $\mathrm{pot}(\mathbb{P}_{t+1}) = m(U_t \cup \{v\}) > m(U_t) = \mathrm{pot}(\mathbb{P}_t)$. And we do not know that the inequality $\mathrm{pot}(\mathbb{P}_t) \leq 2 \cdot \mathit{cost}(\mathbb{P}_t)$ holds, only that $\mathrm{pot}(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathit{cost}(\mathbb{P}_s)\}$. This means that it can happen that $\mathrm{pot}(\mathbb{P}_{t+1}) > 2 \cdot \mathit{cost}(\mathbb{P}_{t+1})$, in which case the induction step fails. However, we claim that using the limited hiding-cardinality property (Property 7.11) we can prove for $U_{t+1} = U_t \cup \{v\}$ that

$$m(U_{t+1}) = m(U_t \cup \{v\}) \leq \max\{m(U_t), 2 \cdot \mathit{cost}(\mathbb{P}_{t+1})\}, \tag{7.2}$$

which shows that (7.1) holds and the induction steps goes through.

Namely, suppose that U_t is chosen as in Property 7.11 and consider $U_{t+1} = U_t \cup \{v\}$. Then U_{t+1} is a hiding set for $\mathbb{P}_{t+1} = (B_t \cup \{v\}, W_t)$ and hence $\mathrm{pot}(\mathbb{P}_{t+1}) \leq m(U_{t+1})$. For $j > 0$, it holds that $U_{t+1}\{\succeq j\} = U_t\{\succeq j\}$ and thus $m^j(U_{t+1}) = m^j(U_t)$. On the bottom level, using that the inequality $|U_t| \leq \mathit{cost}(\mathbb{P}_t)$ holds by the LHC property, we have

$$m^0(U_{t+1}) = 2 \cdot |U_{t+1}| = 2 \cdot (|U_t| + 1) \leq 2 \cdot (\mathit{cost}(\mathbb{P}_t) + 1) = 2 \cdot \mathit{cost}(\mathbb{P}_{t+1}) \tag{7.3}$$

and we get that

$$\begin{aligned} m(U_{t+1}) = \max_j\left\{m^j(U_{t+1})\right\} &= \max\left\{\max_{j>0}\left\{m^j(U_t)\right\}, m^0(U_{t+1})\right\} \\ &\leq \max\{m(U_t), 2 \cdot \mathit{cost}(\mathbb{P}_{t+1})\} = \max\{\mathrm{pot}(\mathbb{P}_t), 2 \cdot \mathit{cost}(\mathbb{P}_{t+1})\} \end{aligned} \tag{7.4}$$

which is exactly what we need.

We see that the inequality (7.1) holds in all cases in our case analysis, which proves the lemma. □

The lower bound on black-white pebbling price now follows by showing that the final pebble configuration $(\{z\}, \emptyset)$ has high potential.

Lemma 7.13. *For z the sink of a pyramid Π_h of height h, the pebble configuration $(\{z\}, \emptyset)$ has potential $\mathrm{pot}_{\Pi_h}(\{z\}, \emptyset) = h + 2$.*

Proof. This follows easily from the limited hiding-cardinality property (which says that U can be chosen so that either $U \subseteq \{z\}$ or $|U| \leq 0$), but let us show that this assumption is not necessary here. The set $U = \{z\}$ hides itself and has measure $m(U) = m^h(U) = h + 2 \cdot 1 = h + 2$. Suppose that z is hidden by some $U' \neq \{z\}$. Without loss of generality U' is minimal, i. e., no strict subset of U' hides z. Let u be a vertex in U' on minimal level $\mathrm{minlevel}(U) = L < h$. The fact that U' is minimal implies that there is a path $P : u \rightsquigarrow z$ such that $(P \setminus \{u\}) \cap U' = \emptyset$ (otherwise $U' \setminus \{u\}$ would hide z). By Observation 7.5, there must exist $h - L$ converging paths from sources to z that are all blocked by distinct pebbles in $U' \setminus \{u\}$. It follows that

$$m(U') \geq m^L(U') = L + 2\left|U'\{\succeq L\}\right| = L + 2\left|U'\right| \geq L + 2 \cdot (h + 1 - L) > h + 2 \tag{7.5}$$

(where we used that $U'\{\succeq L\} = U'$ since $L = \mathrm{minlevel}(U)$). Thus $U = \{z\}$ is the unique minimum-measure hiding set for $(\{z\}, \emptyset)$, and the potential is $\mathrm{pot}(\{z\}, \emptyset) = h + 2$. □

Since [42] proves that pyramids possess the limited hiding-cardinality property, and since there are pebblings that yield matching upper bounds, we have the following theorem.

Theorem 7.14 ([42])**.** $\mathit{BW\text{-}Peb}(\Pi_h) = h/2 + \mathrm{O}(1)$.

Proof. The upper bound was shown in Lemma 7.2. For the lower bound, Lemma 7.13 says that the final pebble configuration $(\{z\}, \emptyset)$ in any complete pebbling $\mathcal{P}$ of Π_h has potential $\mathrm{pot}(\{z\}, \emptyset) = h + 2$. According to Lemma 7.12, $\mathrm{pot}(\{z\}, \emptyset) \leq 2 \cdot \mathit{cost}(\mathcal{P})$. Thus $\mathit{BW\text{-}Peb}(\Pi_h) \geq h/2 + 1$. □

In the remainder of this section, we provide a fairly detailed overview of the proof that pyramids do indeed possess the limited hiding-cardinality property. As was discussed above, the reason for giving all the details is that we will need to use and modify the construction in non-trivial ways in the next section, where we will use ideas inspired by Klawe's paper to prove lower bounds on the pebbling price of pyramids in the blob-pebble game.

7.3 An exposition of the proof of the limited hiding-cardinality property

We present the proof of that pyramids have the limited hiding-cardinality property in a top-down fashion as follows.

1. First, we study what hiding sets look like in order to better understand their structure. Along the way, we make a few definitions and prove some lemmas culminating in Definition 7.20 and Lemma 7.24.

2. We conclude that it seems like a good idea to try to split our hiding set into disjoint components, prove the LHC property locally, and then add everything together to get a proof that works globally. We make an attempt to do this in Theorem 7.25, but note that the argument does not quite work. However, if we assume a slightly stronger property locally for our disjoint components (Property 7.27), the proof goes through.

3. We then establish this stronger local property by assuming that pyramid graphs have a certain *spreading* property (Definition 7.34 and Theorem 7.35) and showing in Lemmas 7.33 and 7.36 that the stronger local property holds for such spreading graphs.

4. To clinch the proof, we also need to show that pyramids are indeed spreading. Since we can use this part from [42] as-is, we do not present a proof here, but instead refer to the original paper or to [48] for a simplified, more direct argument.

For a start, we need two definitions. The intuition for the first one is that the vertex set U is *tight* if it does not contain any "unnecessary" vertex u hidden by the other vertices in U.

Definition 7.15 (Tight vertex set)**.** The vertex set U is *tight* if for all $u \in U$ it holds that $u \notin \llbracket U \setminus \{u\} \rrbracket$.

If x is a vertex hidden by U, we can identify a subset of U that is necessary for hiding x.

Definition 7.16 (Necessary hiding subset)**.** If $x \in \llbracket U \rrbracket$, we define $U_{\llbracket x \rrbracket}$ to be the subset of U such that for each $u \in U_{\llbracket x \rrbracket}$ there is a source path P ending in x for which $P \cap U = \{u\}$.

We observe that if U is tight and $x \in U$, then $U\llbracket x \rrbracket = \{x\}$. This is not the case for non-tight sets. If we let $U = \{x\} \cup pred(x)$ for some non-source x, Definition 7.16 yields that $U\llbracket x \rrbracket = \emptyset$. The vertices in $U\llbracket x \rrbracket$ must be contained in every subset of U that hides x, since for each $v \in U\llbracket x \rrbracket$ there is a source path to x that intersects U only in v. But if U is tight, the set $U\llbracket x \rrbracket$ is also *sufficient* to hide x, i. e., $x \in \lceil\!\lceil U\llbracket x \rrbracket \rceil\!\rceil$.

Lemma 7.17 (Lemma 3.1 in [42])**.** *If U is tight and $x \in \lceil\!\lceil U \rceil\!\rceil$, then $U\llbracket x \rrbracket$ hides x and this set is also contained in every subset of U that hides x.*

Proof. The necessity was argued above, so the interesting part is that $x \in \lceil\!\lceil U\llbracket x \rrbracket \rceil\!\rceil$. Suppose not. Let P_1 be a source path to x such that $P_1 \cap U\llbracket x \rrbracket = \emptyset$. Since U hides x, U blocks P_1. Let v be the highest-level element in $P_1 \cap U$ (i. e., the vertex on this path closest to x). Since U is tight, $U \setminus \{v\}$ does not hide v. Let P_2 be a source path to v such that $P_2 \cap (U \setminus \{v\}) = \emptyset$. Then going first along P_2 and switching to P_1 at v we get a path to x that intersects U only in v. But if so, we have $v \in U\llbracket x \rrbracket$ contrary to assumption. Thus, $x \in \lceil\!\lceil U\llbracket x \rrbracket \rceil\!\rceil$ must hold. □

Given a vertex set U, the tight subset of U hiding the same elements is uniquely determined.

Lemma 7.18. *For any vertex set U in a layered graph G there is a uniquely determined minimal subset $U^* \subseteq U$ such that $\lceil\!\lceil U^* \rceil\!\rceil = \lceil\!\lceil U \rceil\!\rceil$, U^* is tight, and for any $U' \subseteq U$ with $\lceil\!\lceil U' \rceil\!\rceil = \lceil\!\lceil U \rceil\!\rceil$ it holds that $U^* \subseteq U'$.*

Proof. We construct the set U^* bottom-up, layer by layer. We will let U_i^* be the set of vertices on level i or lower in the tight hiding set under construction, and U_i^r be the set of vertices in U strictly above level i remaining to be hidden.

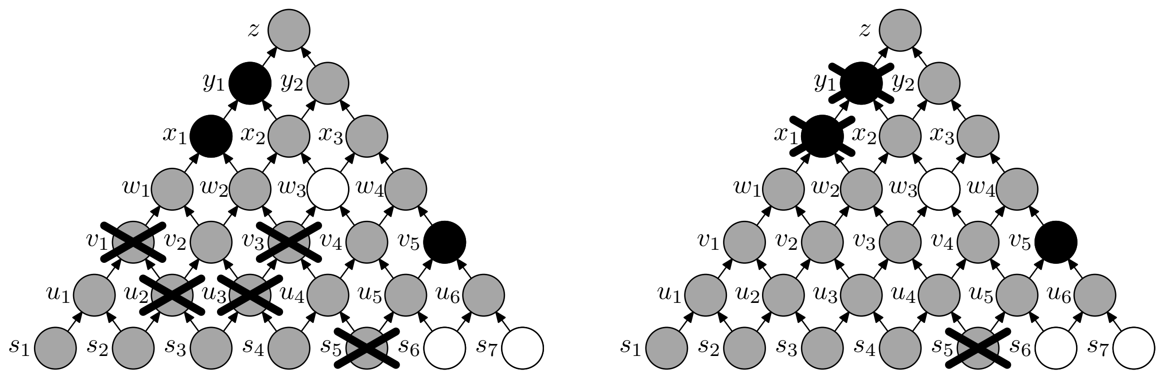
Let $L = \text{minlevel}(U)$. For $i < L$, we define $U_i^* = \emptyset$. Clearly, all vertices on level L in U must be present also in U^*, since no vertices in $U\{\succ L\}$ can hide these vertices and vertices on the same level cannot help hiding each other. Set $U_L^* = U\{\sim L\} = U \setminus U\{\succ L\}$. Now we can remove from U all vertices hidden by U_L^*, so set $U_L^r = U \setminus \lceil\!\lceil U_L^* \rceil\!\rceil$. Note that there are no vertices on or below level L left in U_L^r, i. e., $U_L^r = U_L^r\{\succ L\}$, and that U_L^* hides the same vertices as does $U\{\preceq L\}$ (since the two sets are equal).

Inductively, suppose we have constructed the vertex sets U_{i-1}^* and U_{i-1}^r. Just as above, set $U_i^* = U_{i-1}^* \cup U_{i-1}^r\{\sim i\}$ and $U_i^r = U_{i-1}^r \setminus \lceil\!\lceil U_i^* \rceil\!\rceil$. If there are no vertices remaining on level i to be hidden, i. e., if $U_{i-1}^r\{\sim i\} = \emptyset$, nothing happens and we get $U_i^* = U_{i-1}^*$ and $U_i^r = U_{i-1}^r$. Otherwise the vertices on level i in U_{i-1}^r are added to U_i^* and all of these vertices, as well as any vertices above in U_{i-1}^r now being hidden, are removed from U_{i-1}^r resulting in a smaller set U_i^r.

To conclude, we set $U^* = U_M^*$ for $M = \text{maxlevel}(U)$. By construction, the invariant

$$\lceil\!\lceil U_i^* \rceil\!\rceil = \lceil\!\lceil U\{\preceq i\} \rceil\!\rceil \tag{7.6}$$

holds for all levels i. Thus, $\lceil\!\lceil U^* \rceil\!\rceil = \lceil\!\lceil U \rceil\!\rceil$. Also, U^* must be tight since if $v \in U^*$ and $\text{level}(v) = i$, by construction $U^*\{\prec i\}$ does not hide v, and (as was argued above) neither does $U^*\{\succeq i\} \setminus \{v\}$. Finally, suppose that $U' \subseteq U$ is a hiding set for U with $U^* \nsubseteq U'$. Consider $v \in U^* \setminus U'$ and suppose $\text{level}(v) = i$. On the one hand, we have $v \notin \lceil\!\lceil U_{i-1}^* \rceil\!\rceil$ by construction. On the other hand, by assumption it holds that $v \in \lceil\!\lceil U'\{\prec i\} \rceil\!\rceil$ and thus $v \in \lceil\!\lceil U\{\prec i\} \rceil\!\rceil$. But then by the invariant (7.6) we know that $v \in \lceil\!\lceil U_{i-1}^* \rceil\!\rceil$, which yields a contradiction. Hence, $U^* \subseteq U'$ and the lemma follows. □



(a) Hiding set U with large size and measure.

(b) Smaller hiding set U^* with smaller measure.

Figure 10: Illustration of hiding sets in Example 7.19 (with vertices in hiding sets cross-marked).

We remark that U^* can in fact be seen to contain exactly those elements $u \in U$ such that u is not hidden by $U \setminus \{u\}$.

It follows from Lemma 7.18 that if U is a minimum-measure hiding set for $\mathbb{P} = (B,W)$, we can assume without loss of generality that $U \cup W$ is tight. More formally, if $U \cup W$ is not tight, we can consider minimal subsets $U' \subseteq U$ and $W' \subseteq W$ such that $U' \cup W'$ hides B and is tight, and prove the LHC property for B and W' with respect to this U' instead. Then clearly the LHC property holds also for B and W.

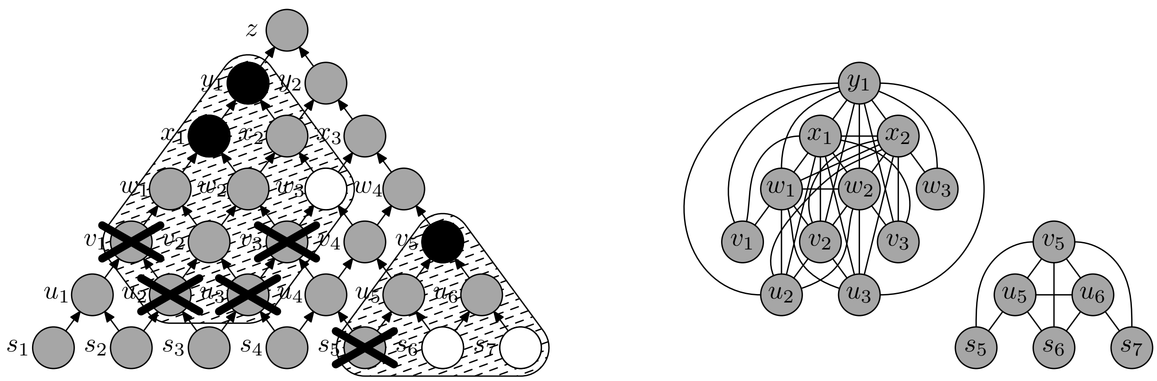
Suppose that we have a set U that together with W hides B. Suppose furthermore that B contains vertices very far apart in the graph. Then it might very well be the case that $U \cup W$ can be split into a number of disjoint subsets $U_i \cup W_i$ responsible for hiding different parts B_i of B, but which are wholly independent of one another. Let us give an example of this.

Example 7.19. Suppose we have the pebble configuration $(B,W) = (\{x_1, y_1, v_5\}, \{w_3, s_6, s_7\})$ and the hiding set $U = \{v_1, u_2, u_3, v_3, s_5\}$ in Figure 10(a). Then $U \cup W$ hides B, but U seems unnecessarily large. To get a better hiding set U^*, we can leave s_5 responsible for hiding v_5 but replace $\{v_1, u_2, u_3, v_3\}$ by $\{x_1, y_1\}$. The resulting set $U^* = \{x_1, y_1, s_5\}$ in Figure 10(b) has both smaller size and smaller measure (we leave the straightforward verification of this fact to the reader).

Intuitively, it seems that the configuration can be split in two components, namely $(B_1, W_1) = (\{x_1, y_1\}, \{w_3\})$ with hiding set $U_1 = \{v_1, u_2, u_3, v_3\}$ and $(B_2, W_2) = (\{v_5\}, \{s_6, s_7\})$ with hiding set $U_2 = \{s_5\}$, and that these two components are independent of one another. To improve the hiding set U, we need to do something locally about the bad hiding set U_1 in the first component, namely replace it with $U_1^* = \{x_1, y_1\}$, but we should keep the locally optimal hiding set U_2 in the second component.

We want to formalize this understanding of how vertices in B, W and U depend on one another in a hiding set $U \cup W$ for B. The following definition constructs a graph that describes the structure of the hiding sets that we are studying in terms of these dependencies.

Definition 7.20 (Hiding set graph)**.** For a tight (and non-empty) set of vertices X in G, the *hiding set graph* $\mathcal{H} = \mathcal{H}(G,X)$ is an undirected graph defined as follows:



(a) Vertices hidden by $U \cup W$. (b) Hiding set graph $\mathcal{H}(U \cup W)$.

Figure 11: Pebble configuration with hiding set and corresponding hiding set graph.

- The set of vertices of $\mathcal{H}$ is $V(\mathcal{H}) = \llbracket X \rrbracket$.
- The set of edges $E(\mathcal{H})$ of $\mathcal{H}$ consists of all pairs of vertices (x,y) for $x,y \in \llbracket X \rrbracket$ such that $G^x_\triangle \cap \llbracket X\llbracket x \rrbracket \rrbracket \cap G^y_\triangle \cap \llbracket X\llbracket y \rrbracket \rrbracket \neq \emptyset$.

We say that the vertex set X is *hiding-connected* if $\mathcal{H}(G,X)$ is a connected graph.

When the graph G and vertex set X are clear from context, we will sometimes write only $\mathcal{H}(X)$ or even just $\mathcal{H}$. To illustrate Definition 7.20, we give an example.

Example 7.21. Consider again the configuration $(B,W) = (\{x_1,y_1,v_5\},\{w_3,s_6,s_7\})$ from Example 7.19 with hiding set $U = \{v_1,u_2,u_3,v_3,s_5\}$, where we have shaded the set of hidden vertices in Figure 11(a). The hiding set graph $\mathcal{H}(X)$ for $X = U \cup W = \{v_1,u_2,u_3,v_3,w_3,s_5,s_6,s_7\}$ has been drawn in Figure 11(b). In accordance with the intuition sketched in Example 7.19, $\mathcal{H}(X)$ consists of two connected components.

Note that there are edges from the top vertex y_1 in the first component to every other vertex in this component and from the top vertex v_5 to every other vertex in the second component. We will prove presently that this is always the case (Lemma 7.22). Perhaps a more interesting edge in $\mathcal{H}(X)$ is, e. g., (w_1,x_2). This edge exists since $X\llbracket w_1 \rrbracket = \{v_1,u_2,u_3\}$ and $X\llbracket x_2 \rrbracket = \{u_2,u_3,v_3,w_3\}$ intersect and since as a consequence of this (which is easily verified) we have $\Pi^{w_1}_\triangle \cap \llbracket X\llbracket w_1 \rrbracket \rrbracket \cap \Pi^{x_2}_\triangle \cap \llbracket X\llbracket x_2 \rrbracket \rrbracket \neq \emptyset$. For the same reason, there is an edge (u_5,u_6) since $X\llbracket u_5 \rrbracket = \{s_5,s_6\}$ and $X\llbracket u_6 \rrbracket = \{s_6,s_7\}$ intersect.

Lemma 7.22. *Suppose for a tight vertex set X that $x \in \llbracket X \rrbracket$ and $y \in X\llbracket x \rrbracket$. Then x and y are in the same connected component of $\mathcal{H}(X)$.*

Proof. Note first that $x,y \in \llbracket X \rrbracket$ by assumption, so x and y are both vertices in $\mathcal{H}(X)$. Since x is above y we have $G^x_\triangle \supseteq G^y_\triangle$ and we get $G^x_\triangle \cap \llbracket X\llbracket x \rrbracket \rrbracket \cap G^y_\triangle \cap \llbracket X\llbracket y \rrbracket \rrbracket = \llbracket X\llbracket x \rrbracket \rrbracket \cap G^y_\triangle \cap \{y\} = \{y\} \neq \emptyset$. Thus, (x,y) is an edge in $\mathcal{H}(X)$, so x and y are certainly in the same connected component. □

Corollary 7.23. *If X is tight and $x \in \llbracket X \rrbracket$, then x and all of $X\llbracket x \rrbracket$ are in the same connected component of the hiding set graph $\mathcal{H}(X)$.*

The next lemma says that if $\mathcal{H}(X)$ is a hiding set graph with vertex set $V = [\![X]\!]$, then the connected components $V_1,\ldots,V_k$ of $\mathcal{H}(X)$ are themselves hiding set graphs defined over the hiding-connected subsets $X \cap V_1,\ldots,X \cap V_k$.

Lemma 7.24 (Lemma 3.3 in [42])**.** *Let X be a tight set and let V_i be one of the connected components in $\mathcal{H}(X)$. Then the subgraph of $\mathcal{H}(X)$ induced by V_i is identical to the hiding set graph $\mathcal{H}(X \cap V_i)$ defined on the vertex subset $X \cap V_i$. In particular, it holds that $V_i = [\![X \cap V_i]\!]$.*

Proof. We need to show that $V_i = [\![X \cap V_i]\!]$ and that the edges of $\mathcal{H}(X)$ in V_i are exactly the edges in $\mathcal{H}(X \cap V_i)$. Let us first show that $y \in V_i$ if and only if $y \in [\![X \cap V_i]\!]$.

($\Rightarrow$) Suppose $y \in V_i$. Then $X\|y\| \subseteq V_i$ by Corollary 7.23. Also, by definition it holds that $X\|y\| \subseteq X$, so $X\|y\| \subseteq X \cap V_i$. Since $y \in [\![X\|y\|]\!]$ by Lemma 7.17, clearly $y \in [\![X \cap V_i]\!]$.

($\Leftarrow$) Suppose $y \in [\![X \cap V_i]\!]$. Since X is tight, its subset $X \cap V_i$ must be tight as well. Applying Lemma 7.17 twice, we deduce that $(X \cap V_i)\|y\|$ hides y and that $X\|y\| \subseteq (X \cap V_i)\|y\|$ since $X\|y\|$ is contained in any subset of X that hides y. But then a third appeal to Lemma 7.17 yields that $(X \cap V_i)\|y\| \subseteq X\|y\|$ since $X\|y\| \subseteq (X \cap V_i)\|y\| \subseteq X \cap V_i$ and consequently

$$X\|y\| = (X \cap V_i)\|y\| \,. \tag{7.7}$$

By Corollary 7.23, y and all of $(X \cap V_i)\|y\| = X\|y\|$ are in the same connected component. Since $X\|y\| \subseteq V_i$ it follows that $y \in V_i$.

This shows that $V_i = [\![X \cap V_i]\!]$. Plugging (7.7) into Definition 7.20, we see that (x,y) is an edge in $\mathcal{H}(X)$ for $x,y \in V_i$ if and only if (x,y) is an edge in $\mathcal{H}(X \cap V_i)$. □

Now we are in a position to describe the structure of the proof that pyramid graphs have the LHC property.

Theorem 7.25 (Analogue of Theorem 3.7 in [42])**.** *Let $\mathbb{P} = (B,W)$ be any black-white pebble configuration on a pyramid Π. Then there is a vertex set U such that $U \cup W$ hides B, $\mathrm{pot}_{\Pi}(\mathbb{P}) = m(U)$ and either $U = B$ or $|U| < |B| + |W|$.*

The idea is to construct the graph $\mathcal{H} = \mathcal{H}(\Pi, U \cup W)$, study the different connected components in $\mathcal{H}$, find good hiding sets locally that satisfy the LHC property (which we prove is true for each local hiding-connected subset of $U \cup W$), and then add all of these partial hiding sets together to get a globally good hiding set.

Unfortunately, this does not quite work. Let us nevertheless attempt to do the proof, note where and why it fails, and then see how Klawe fixes the broken details.

Tentative proof of Theorem 7.25. Let U be a set of vertices in the pyramid graph Π such that $U \cup W$ hides B and $\mathrm{pot}(\mathbb{P}) = m(U)$. Suppose that U has minimal size among all such sets, and furthermore that among all such minimum-measure and minimum-size sets U has the largest intersection with B.

Assume without loss of generality (Lemma 7.18) that $U \cup W$ is tight, so that we can construct $\mathcal{H}$. Let the connected components of $\mathcal{H}$ be $V_1,\ldots,V_k$. For all $i = 1,\ldots,k$, let $B_i = B \cap V_i$, $W_i = W \cap V_i$, and $U_i = U \cap V_i$. Lemma 7.24 says that $U_i \cup W_i$ hides B_i. In addition, all V_i are pairwise disjoint, so $|B| = \sum_{i=1}^{k} |B_i|$, $|W| = \sum_{i=1}^{k} |W_i|$ and $|U| = \sum_{i=1}^{k} |U_i|$.

Thus, if the LHC property (Property 7.11) does not hold for U globally, there is some hiding-connected subset $U_i \cup W_i$ that hides B_i but for which $|U_i| \geq |B_i| + |W_i|$ and $U_i \neq B_i$. Note that this implies that $B_i \nsubseteq U_i$ since otherwise U_i would not be minimal.

Suppose that we would know that the LHC property is true for each connected component. Then we could find a vertex set U_i^* with $U_i^* \subseteq B_i$ or $\left|U_i^*\right| < |B_i| + |W_i|$ such that $U_i^* \cup W_i$ hides B_i and $m\bigl(U_i^*\bigr) \leq m(U_i)$. Setting $U^* = (U \setminus U_i) \cup U_i^*$, we would get a hiding set with either $|U^*| < |U|$ or $|U^* \cap B| > |U \cap B|$. The second inequality would hold since if $|U^*| = |U|$, then $\left|U_i^*\right| = |U_i| \geq |B_i \cup W_i|$ and this would imply $U_i^* = B_i$ and thus $\left|U_i^* \cap B_i\right| > |U_i \cap B_i|$. This would contradict how U was chosen above, and we would be home.

Almost. We would also need that U_i^* could be substituted for U_i in U without increasing the measure, i. e., that $m\bigl(U_i^*\bigr) \leq m\bigl(U_i\bigr)$ should imply $m\bigl((U \setminus U_i) \cup U_i^*\bigr) \leq m\bigl((U \setminus U_i) \cup U_i\bigr)$. And this turns out not to be true. □

The reason that the proof above does not quite work is that the measure in Definition 7.8 is ill-behaved with respect to unions. Klawe provides the following example of what can happen.

Example 7.26. With vertex labels as in Figures 7 and 9–11, let $X_1 = \{s_1, s_2\}$, $X_2 = \{w_1\}$ and $X_3 = \{s_3\}$. Then $m(X_1) = 4$ and $m(X_2) = 5$ but taking unions with X_3 we get that $m(X_1 \cup X_3) = 6$ and $m(X_2 \cup X_3) = 5$. Thus $m(X_1) < m(X_2)$ but $m(X_1 \cup X_3) > m(X_2 \cup X_3)$.

So it is not enough to show the LHC property locally for each connected component in the graph. We also need that sets U_i from different components can be combined into a global hiding set while maintaining measure inequalities. This leads to the following strengthened condition for connected components of $\mathcal{H}$.

Property 7.27 (Local limited hiding-cardinality property)**.** We say that the pebble configuration $\mathbb{P} = (B, W)$ has the *local limited hiding-cardinality property*, or just the *local LHC property* for short, if for any vertex set U such that $U \cup W$ hides B and is hiding-connected, we can find a vertex set U^* such that

1. U^* is a hiding set for (B, W),
2. for any vertex set Y with $Y \cap U = \emptyset$ it holds that $m\bigl(Y \cup U^*\bigr) \leq m(Y \cup U)$,
3. $U^* \subseteq B$ or $\left|U^*\right| < |B| + |W|$.

We say that the graph G has the local LHC property if all black-white pebble configurations $\mathbb{P} = (B, W)$ on G do.

Note that if the local LHC property holds, this in particular implies that $m\bigl(U^*\bigr) \leq m(U)$ (just choose $Y = \emptyset$). Also, we immediately get that the LHC property holds globally.

Lemma 7.28. *If G has the local limited hiding-cardinality property (Property 7.27), then G has the limited hiding-cardinality property (Property 7.11).*

Proof. Consider the tentative proof of Theorem 7.25 and look at the point where it breaks down. If we instead use the local LHC property to find U_i^*, this time we get that $m\bigl(U_i^*\bigr) \leq m\bigl(U_i\bigr)$ does indeed imply $m\bigl((U \setminus U_i) \cup U_i^*\bigr) \leq m\bigl((U \setminus U_i) \cup U_i\bigr)$, and the theorem follows. □

An obvious way to get the inequality $m(Y \cup U^*) \leq m(Y \cup U)$ in Property 7.27 would be to require that $m^j(U^*) \leq m^j(U)$ for all j, but we need to be slightly more general. The next definition identifies a sufficient condition for sets to behave well under unions with respect to the measure in Definition 7.8.

Definition 7.29. We write $U \precsim_m V$ if for all $j \geq 0$ there is an $i \leq j$ such that $m^j(U) \leq m^i(V)$.

Note that it is sufficient to verify the condition in Definition 7.29 for $j = 1, \ldots, \mathrm{maxlevel}(U)$. For $j > \mathrm{maxlevel}(U)$ we get $m^j(U) = 0$ and the inequality trivially holds.

It is immediate that $U \precsim_m V$ implies $m(U) \leq m(V)$, but the relation $\precsim_m$ gives us more information than that. The usual inequality $m(U) \leq m(V)$ holds if and only if for every j we can find an i such that $m^j(U) \leq m^i(V)$, but in the definition of $\precsim_m$ we are restricted to finding such an index i that is less than or equal to j. So not only is $m(U) \leq m(V)$ globally, but we can also explain locally at each level, by "looking downwards," why U has smaller measure than V.

In Example 7.26, $X_1 \not\precsim_m X_2$ since the relative cheapness of X_1 compared to X_2 is explained not by a lot of vertices in X_2 on low levels, but by one single high-level, and therefore expensive, vertex in X_2 which is far above X_1. This is why these sets behave badly under union. If we have two sets X_1 and X_2 with $X_1 \precsim_m X_2$, however, reversals of measure inequalities when taking unions as in Example 7.26 can no longer occur.

Lemma 7.30 (Lemma 3.4 in [42]). *If $U \precsim_m V$ and $Y \cap V = \emptyset$, then $m(Y \cup U) \leq m(Y \cup V)$.*

Proof. To show that $m(Y \cup U) \leq m(Y \cup V)$, for each level $j = 1, \ldots, \mathrm{maxlevel}(Y \cup U)$ we want to find a level i such that $m^j(Y \cup U) \leq m^i(Y \cup V)$. We pick the $i \leq j$ provided by the definition of $U \precsim_m V$ such that $m^j(U) \leq m^i(V)$. Since $V \cap W = \emptyset$ and $i \leq j$ implies $Y\{\succeq j\} \subseteq Y\{\succeq i\}$, we get

$$\begin{aligned} m^j(Y \cup U) = j + 2 \cdot |(U \cup Y)\{\succeq j\}| \leq j + 2 \cdot |U\{\succeq j\}| + 2 \cdot |Y\{\succeq j\}| \leq \\ i + 2 \cdot |V\{\succeq i\}| + 2 \cdot |Y\{\succeq i\}| = m^i(Y \cup V) \end{aligned} \tag{7.8}$$

and the lemma follows. □

So when locally improving a hiding set U that does not satisfy the LHC property to some set U^* that does, if we can take care that $U^* \precsim_m U$ in the sense of Definition 7.29 we get the local LHC property. All that remains is to show that this can indeed be done.

When "improving" U to U^*, we will strive to pick hiding sets of minimal size. The next definition makes this precise.

Definition 7.31. For any set of vertices X, let

$$L_{\succeq j}(X) = \min\{|Y| : X\{\succeq j\} \subseteq [\![Y]\!] \text{ and } Y\{\succeq j\} = Y\}$$

denote the size of a smallest set Y such that all vertices in Y are on level j or higher and Y hides all vertices in X on level j or higher.

Note that we only require of Y to hide $X\{\succeq j\}$ and not all of X. Given the condition that $Y = Y\{\succeq j\}$, this set cannot hide any vertices in $X\{\prec j\}$. We make a few easy observations.

Observation 7.32. *Suppose that X is a set of vertices in a layered graph G. Then:*

1. *$L_{\succeq 0}(X)$ is the minimal size of any hiding set for X.*
2. *If $X \subseteq Y$, then $L_{\succeq j}(X) \leq L_{\succeq j}(Y)$ for all j.*
3. *It always holds that $L_{\succeq j}(X) \leq |X\{\succeq j\}| \leq |X|$.*

Proof. Part 1 follows from the fact that $V\{\succeq 0\} = V$ for any set V. If $X \subseteq Y$, then $X\{\succeq j\} \subseteq Y\{\succeq j\}$ and any hiding set for $X\{\succeq j\}$ works also for $Y\{\succeq j\}$, which yields part 2. Part 3 holds since $X\{\succeq j\} \subseteq X$ is always a possible hiding set for itself. □

For any vertex set V in any layered graph G, we can always find a set hiding V that has "minimal cardinality at each level" in the sense of Definition 7.31.

Lemma 7.33 (Lemma 3.5 in [42])**.** *For any vertex set V we can find a hiding set V^* such that $\left|V^*\{\succeq j\}\right| \leq L_{\succeq j}(V)$ for all j, and either $V^* = V$ or $|V^*| < |V|$.*

Proof. If $|V\{\succeq j\}| \leq L_{\succeq j}(V)$ for all j, we can choose $V^* = V$. Suppose this is not the case, and let k be minimal such that $|V\{\succeq k\}| > L_{\succeq k}(V)$. Let V' be a minimum-size hiding set for $V\{\succeq k\}$ with $V' = V'\{\succeq k\}$ and $\left|V'\right| = |L_{\succeq k}(V)|$ and set $V^* = V\{\prec k\} \dot\cup V'$. Since $V\{\prec k\}$ hides itself (any set does), we have that V^* hides $V = V\{\prec k\} \dot\cup V\{\succeq k\}$ and that

$$\left|V^*\right| = |V\{\prec k\}| + |V'| < |V\{\prec k\}| + |V\{\succeq k\}| = |V|\,. \tag{7.9}$$

Combining (7.9) with part 1 of Observation 7.32, we see that the minimal index found above must be $k = 0$. Going through the same argument as above again, we see that $\left|V^*\{\succeq j\}\right| \leq L_{\succeq j}(V)$ for all j, since otherwise (7.9) would yield a contradiction to the fact that $V' = V'\{\succeq 0\}$ was chosen as a minimum-size hiding set for V. □

We noted above that $L_{\succeq 0}(X)$ is the cardinality of a minimum-size hiding set of X. For $j > 0$, the quantity $L_{\succeq j}(X)$ is large if one needs many vertices on level $\geq j$ to hide $X\{\succeq j\}$, i. e., if $X\{\succeq j\}$ is "spread out" in some sense. Let us consider a pyramid graph and suppose that X is a tight and hiding-connected set in which the level-difference $\mathrm{maxlevel}(X) - \mathrm{minlevel}(X)$ is large. Then it seems that $|X|$ should also have to be large, since the pyramid "fans out" so quickly. This intuition might be helpful when looking at the next, crucial definition of Klawe.

Definition 7.34 (Spreading graph)**.** We say that the layered DAG G is a *spreading graph* if for every (non-empty) hiding-connected set X in G and every level $j = 1, \ldots, \mathrm{maxlevel}(\llbracket X \rrbracket)$, the *spreading inequality*

$$|X| \geq L_{\succeq j}(\llbracket X \rrbracket) + j - \mathrm{minlevel}(X) \tag{7.10}$$

holds.

Let us try to give some more intuition for Definition 7.34 by considering two extreme cases in a pyramid graph:

- For $j \leq \text{minlevel}(X)$, we have that the term $j - \text{minlevel}(X)$ is non-positive, $X\{\succeq j\} = X$, and $\llbracket X\{\succeq j\} \rrbracket = \llbracket X \rrbracket$. In this case, (7.10) is just the trivial fact that no set that hides $\llbracket X \rrbracket$ need be larger than X itself.

- Consider $j = \text{maxlevel}(\llbracket X \rrbracket)$, and suppose that $\llbracket X\{\succeq j\} \rrbracket$ is a single vertex x with $X_{\llbracket x \rrbracket} = X$. Then (7.10) requires that $|X| \geq 1 + \text{level}(x) - \text{minlevel}(X)$, and this can be proven to hold by the "converging paths" argument of Theorem 7.3 and Observation 7.5.

Very loosely, Definition 7.34 says that if X contains vertices at low levels that help to hide other vertices at high levels, then X must be a large set. Just as we tried to argue above, the spreading inequality (7.10) does indeed hold for pyramids.

Theorem 7.35 ([42]). *Pyramids are spreading graphs.*

Unfortunately, the proof of Theorem 7.35 in [42] is rather involved. The analysis is divided into two parts, by first showing that a class of so-called *nice graphs* are spreading, and then demonstrating that pyramid graphs are nice. We refer to the original paper for the details. An alternative proof, which uses a more direct argument focusing specifically on pyramids and which might be of independent interest, can be found in [48].

Accepting Theorem 7.35 on faith, we are ready for the decisive lemma: If our layered DAG is a spreading graph and if $U \cup W$ is a hiding-connected set hiding B such that U is too large for the conditions in the local limited hiding-cardinality property (Property 7.27) to hold, then replacing U by the minimum-size hiding set in Lemma 7.33 we get a hiding set in accordance with the local LHC property.

Lemma 7.36 (Lemma 3.6 in [42]). *Suppose that B, W, U are vertex sets in a layered spreading graph G such that $U \cup W$ hides B and is tight and hiding-connected. Then there is a vertex set U^* such that $U^* \cup W$ hides B, $U^* \precsim_m U$, and either $U^* = B$ or $|U^*| < |B| + |W|$.*

Postponing the proof of Lemma 7.36 for a moment, let us note that if we combine this lemma with Lemma 7.30 and Theorem 7.35, the local limited hiding-cardinality property for pyramids follows.

Corollary 7.37. *Pyramid graphs have the local limited hiding-cardinality property (Property 7.27).*

Proof of Corollary 7.37. This is more or less immediate, but we write down the details for completeness. Since pyramids are spreading by Theorem 7.35, Lemma 7.36 says that U^* is a hiding set for (B, W) and that $U^* \precsim_m U$. Lemma 7.30 then yields that $m(Y \cup U^*) \leq m(Y \cup U)$ for all Y with $Y \cap U = \emptyset$. Finally, Lemma 7.36 also tells us that $U^* \subseteq B$ or $|U^*| < |B| + |W|$, and thus all conditions in Property 7.27 are satisfied. □

Continuing by plugging Corollary 7.37 into Lemma 7.28, we get the global LHC property in Theorem 7.25 on page 520. So all that is needed to conclude Klawe's proof of the lower bound for the black-white pebbling price of pyramids is to prove Theorem 7.35 and Lemma 7.36. We attend to Lemma 7.36 right away but in order to save space we skip the proof of Theorem 7.35 here and instead refer the reader to the exposition in [48].

Proof of Lemma 7.36. If $|U| < |B| + |W|$ we can pick $U^* = U$ and be done, so suppose that $|U| \geq |B| + |W|$. Intuitively, this should mean that U is unnecessarily large, so it ought to be possible to do better. In fact, U is so large that we can just ignore W and pick a better U^* that hides B all on its own.

Namely, let U^* be a minimum-size hiding set for B as in Lemma 7.33. Then either $U^* = B$ or $\left|U^*\right| < |B| \leq |B| + |W|$. To prove the lemma, we also need to show that $U^* \precsim_m U$, which will guarantee that U^* behaves well under union with other sets with respect to measure.

Before we do the the formal calculations, let us try to provide some intuition for why it should be the case that $U^* \precsim_m U$ holds, i. e., that for every j we can find an $i \leq j$ such that $m^j\left(U^*\right) \leq m^i(U)$. Perhaps it will be helpful at this point for the reader to look at Example 7.19 again, where the replacement of $U_1 = \{v_1, u_2, u_3, v_3\}$ in Figure 10(a) by $U_1^* = \{x_1, y_1\}$ in Figure 10(b) shows Lemmas 7.33 and 7.36 in action.

Suppose first that $j \leq \mathrm{minlevel}(U \cup W) \leq \mathrm{minlevel}(U)$. Then the inequality $m^j(U^*) \leq m^j(U)$ is obvious, since $U\{\succeq j\} = U$ is so large that it can easily pay for all of U^*, let alone $U^*\{\succeq j\} \subseteq U^*$.

For $j > \mathrm{minlevel}(U \cup W)$, however, we can worry that although our hiding set U^* does indeed have small size, the vertices in U^* might be located on high levels in the graph and be very expensive since they were chosen without regard to measure. Just throwing away all white pebbles and picking a new set U^* that hides B on its own is quite a drastic move, and it is not hard to construct examples where this is very bad in terms of potential (say, exchanging s_5 for v_5 in the hiding set of Example 7.19). The reason that this nevertheless works is that $|U|$ is so large, that, in addition, $U \cup W$ is hiding-connected, and that, finally, the graph under consideration is spreading. Thanks to this, if there are a lot of expensive vertices in $U^*\{\succeq j\}$ on or above some high level j resulting in a large partial measure $m^j\left(U^*\right)$, the number of vertices on or above level $L = \mathrm{minlevel}(U \cup W)$ in $U = U\{\succeq L\}$ is large enough to yield at least as large a partial measure $m^L\left(U\right)$.

Let us do the formal proof, divided into the two cases above.

1. $j \leq \mathrm{minlevel}(U \cup W)$: Using the lower bound on the size of U and that level j is no higher than the minimal level of U, we get

$$
\begin{aligned}
m^j\left(U^*\right) &= j + 2 \cdot \left|U^*\{\succeq j\}\right| && \left[\text{ by definition of } m^j(\cdot) \right] \\
&\leq j + 2 \cdot \left|U^*\right| && \left[\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V \right] \\
&\leq j + 2 \cdot |B| && \left[\text{ by construction of } U^* \text{ in Lemma 7.33} \right] \\
&\leq j + 2 \cdot |U| && \left[\text{ by assumption } |U| \geq |B| + |W| \geq |B| \right] \\
&= j + 2 \cdot \left|U\{\succeq j\}\right| && \left[\ U\{\succeq j\} = U \text{ since } j \leq \mathrm{minlevel}(U) \right] \\
&= m^j(U) && \left[\text{ by definition of } m^j(\cdot) \right]
\end{aligned}
$$

and we can choose $i = j$ in Definition 7.29.

2. $j > \mathrm{minlevel}(U \cup W)$: Let $L = \mathrm{minlevel}(U \cup W)$. The black pebbles in B are hidden by $U \cup W$, or in formal notation $B \subseteq \llbracket U \cup W \rrbracket$, so

$$
L_{\succeq j}(B) \leq L_{\succeq j}\left(\llbracket U \cup W \rrbracket\right) \tag{7.11}
$$

holds by part 2 of Observation 7.32. Moreover, $U \cup W$ is a hiding-connected set of vertices in a spreading graph G, so the spreading inequality in Definition 7.34 says that $|U \cup W| \geq$

$L_{\succeq j}\big(\llbracket U \cup W \rrbracket\big) + j - L$, or

$$j + L_{\succeq j}\big(\llbracket U \cup W \rrbracket\big) \leq L + |U \cup W| \tag{7.12}$$

after reordering. Combining (7.11) and (7.12) we have that

$$j + L_{\succeq j}(B) \leq L + |U \cup W| \tag{7.13}$$

and it follows that

$$\begin{aligned}
m^j(U^*) &= j + 2 \cdot \left|U^*\{\succeq j\}\right| && \big[\text{ by definition of } m^j(\cdot) \ \big] \\
&\leq j + \left|U^*\{\succeq j\}\right| + \left|U^*\right| && \big[\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V \ \big] \\
&\leq j + L_{\succeq j}(B) + |B| && \big[\text{ by construction of } U^* \text{ in Lemma 7.33 } \big] \\
&\leq L + |U \cup W| + |B| && \big[\text{ by the inequality (7.13) } \big] \\
&\leq L + 2 \cdot |U| && \big[\text{ by assumption } |U| \geq |B| + |W| \ \big] \\
&= L + 2 \cdot |U\{\succeq L\}| && \big[\ U\{\succeq L\} = U \text{ since } L \leq \text{minlevel}(U) \ \big] \\
&= m^L(U) && \big[\text{ by definition of } m^L(\cdot) \ \big] \,.
\end{aligned}$$

Thus, the partial measure of U at the minimum level L is always larger than the partial measure of U^* at levels j above this minimum level, and we can choose $i = L$ in Definition 7.29.

Consequently, $U^* \precsim_m U$, and the lemma follows. □

Thereby, the proof of the lower bound on the black-white pebbling price of pyramid graphs in Theorem 7.14 on page 516 is complete.

Concluding this subsection, we want to make a comment about Lemmas 7.33 and 7.36 and try to rephrase what they say about hiding sets. Given a tight set $U \cup W$ such that $B \subseteq \llbracket U \cup W \rrbracket$, we can always pick a U^* as in Lemma 7.33 with $U^* = B$ or $\left|U^*\right| < |B| + |W|$ and with $\left|U^*\{\succeq j\}\right| \leq L_{\succeq j}(B)$ for all j. This will sometimes be a good idea, and sometimes not. Just as in Lemma 7.36, for $j > \text{minlevel}(U \cup W)$ we can always prove that

$$m^j(U^*) \leq \text{minlevel}(U \cup W) + |U| + (|B| + |W|)\,. \tag{7.14}$$

The key message of Lemma 7.36 is that replacing U by U^* is a good idea if U is sufficiently large, namely if $|U| \geq |B| + |W|$, in which case we are guaranteed to get $m^j(U^*) \leq m^L(U)$ for $L = \text{minlevel}(U \cup W)$.

8 A tight bound for blob-pebbling the pyramid

Inspired by Klawe's ideas in the previous section for the standard black-white pebble game, we want to prove a similar result for our new blob-pebble game. In this section, we do so by showing that for all layered DAGs G_h of height h that are spreading (in the sense of Definition 7.34), it holds that *Blob-Peb*$(G_h) = \Theta(h)$. In particular, this bound holds for pyramids Π_h.

The constant factor that we get in our lower bound is moderately small and explicit. In fact, we believe that it should hold that *Blob-Peb*$(G_h) \geq h/2 + \mathrm{O}(1)$ for layered spreading graphs G_h of height h, just as in the standard black-white pebble game, but we have not been able to prove this. The multiplicative constant in the proof in this section can improved with a minor effort, but we have not made any real attempt to optimize the calculations since it seems clear that some additional idea would be needed to push the constant all the way up to $1/2$.

8.1 Definitions and notation for the blob-pebbling price lower bound

Recall that a vertex set U hides a black pebble on b if it blocks all source paths visiting v. We want to extend this definition to blobs B. While this might not be a priori entirely obvious, it turns out to be fruitful to do so by requiring that U should block all paths going through all of B. We remind the reader about the terminology and notation from Definition 4.4 that a black blob B and a path P *agree* with each other, or that P is a path *via* B, if $B \subseteq P$, and that $\mathfrak{P}_{\mathrm{via}}(B)$ denotes the set of all source paths agreeing with B.

Definition 8.1 (Blocked black blob)**.** A vertex set U *blocks* a blob B if U blocks all $P \in \mathfrak{P}_{\mathrm{via}}(B)$.

Note, in particular, that this means that a blob B that is *not* a chain is vacuously blocked even by the empty set, since $\mathfrak{P}_{\mathrm{via}}(B) = \emptyset$ in this case (i. e., there is no path agreeing with B). Again, while we do not claim that this is obvious, when working with the blob-pebble game one can actually develop a certain intuition why non-chain blobs should be fairly useless. Definition 8.1 captures this intuition by specifying that there will be no need to block this kind of blobs when we lift Klawe's proof to the blob-pebble game.[5]

A terminological aside: Recalling the discussion in the beginning of Section 7.2, it seems natural to say that U *blocks* a black blob B rather than hides it, since standing at the sources we might "see" the beginning of B, but if we try to walk any path via B we will fail before reaching the top of B since U blocks the path. This distinction between hiding and blocking will turn out to be an important one in our lower bound proof for blob-pebbling price. Of course, if B is an atomic black pebble, i. e., $|B| = 1$, the hiding and blocking relations coincide.

Let us next define what it means to block a blob-pebbling configuration.

Definition 8.2 (Unblocked paths)**.** For $[B]\langle W\rangle$ an blob subconfiguration, the set of *unblocked paths* for $[B]\langle W\rangle$ is

$$\mathrm{unblocked}([B]\langle W\rangle) = \{P \in \mathfrak{P}_{\mathrm{via}}(B) \mid W \text{ does not block } P\}$$

and we say that U blocks $[B]\langle W\rangle$ if U blocks all paths in $\mathrm{unblocked}([B]\langle W\rangle)$. We say that U blocks the blob-pebbling configuration $\mathbb{S}$ if U blocks all $[B]\langle W\rangle \in \mathbb{S}$. If so, we say that U is a *blocker* of $[B]\langle W\rangle$ or $\mathbb{S}$, respectively, or a *blocking set* for $[B]\langle W\rangle$ or $\mathbb{S}$.

Comparing to Section 7.2, note that when blocking a path $P \in \mathfrak{P}_{\mathrm{via}}(B)$, U can only use the white pebbles W that are associated with B in $[B]\langle W\rangle$. Although there might be white pebbles from other subconfigurations $[B']\langle W'\rangle \neq [B]\langle W\rangle$ that would be really helpful, U cannot enlist the help of the white pebbles

[5] In fact, in the preliminary version [50] of this paper, black blobs were declared to be chains by definition. However, while this was, and is, in some sense intuitively more appealing, it turned out to lead to a number of technical difficulties that are more easily overcome or even entirely avoided by letting blobs span any set of vertices, chains or not.

in W' when blocking B. The reason for defining the blocking relation in this way is that these white pebbles can suddenly disappear due to pebbling moves performed on such subconfigurations $[B']\langle W'\rangle$.

Reusing the definition of measure in Definition 7.8 on page 512, we generalize the concept of potential in Definition 7.9 to blob-pebbling configurations as follows.

Definition 8.3 (Blob-pebbling potential)**.** The *potential* of a blob-pebbling configuration $\mathbb{S}$ is

$$\mathrm{pot}(\mathbb{S}) = \min\{m(U) : U \text{ blocks } \mathbb{S}\}.$$

If U is such that U blocks $\mathbb{S}$ and U has minimal measure $m(U)$ among all blocking sets for $\mathbb{S}$, we say that U is a *minimum-measure* blocking set for $\mathbb{S}$.

To compare blob-pebbling potential with the black-white pebbling potential in Definition 7.9, consider the following examples with vertex labels as in Figures 7 and 9–11.

Example 8.4. For the blob-pebbling configuration $\mathbb{S} = \big\{[z]\langle y_1\rangle, [z]\langle y_2\rangle\big\}$, the minimum-measure blocking set is $U = \{z\}$. In comparison, the standard black-white pebble configuration $\mathbb{P} = (B,W) = (\{z\},\{y_1,y_2\})$ has $U = \emptyset$ as minimum-measure hiding set.

Example 8.5. For the blob-pebbling configuration $\mathbb{S} = \big\{[z]\langle\emptyset\rangle, [y_1]\langle x_1,x_2\rangle\big\}$, the minimum-measure blocker is again $U = \{z\}$. In comparison, for the standard black-white pebble configuration $\mathbb{P} = (B,W) = (\{z,y_1\},\{x_1,x_2\})$ we have the minimum-measure hiding set $U = \{x_3\}$.

Remark 8.6. Perhaps it is also worth pointing out that Definition 8.3 is indeed a strict generalization of Definition 7.9. Given a black-white pebble configuration $\mathbb{P} = (B,W)$ we can construct an equivalent blob-pebbling configuration $\mathbb{S}(\mathbb{P})$ with respect to potential by setting

$$\mathbb{S}(\mathbb{P}) = \big\{ [b]\big\langle W \cap G^b_\triangle \big\rangle \big| b \in B \big\} \tag{8.1}$$

but as the examples above show going in the other direction is not possible.

Since we have accumulated a number of different minimality criteria for blocking sets, let us pause to clarify the terminology:

- The vertex set U is a *subset-minimal*, or just *minimal*, blocking set for the blob-pebbling configuration $\mathbb{S}$ if no strict subset $U' \subsetneq U$ is a blocking set for $\mathbb{S}$.
- U is a *minimum-measure* blocking set for $\mathbb{S}$ if it has minimal measure among all blocking sets for $\mathbb{S}$ (and thus yields the potential of $\mathbb{S}$).
- U is a *minimum-size* blocking set for $\mathbb{S}$ if it has minimal size among all blocking sets for $\mathbb{S}$.

Note that we can assume without loss of generality that minimum-measure and minimum-size blockers are both subset-minimal, since throwing away superfluous vertices can only decrease the measure and size, respectively. However, minimum-measure blockers need not have minimal size and vice versa. For a simple example of this, consider (with vertex labels as in Figures 7 and 9–11) the blob-pebbling configuration $\mathbb{S} = \big\{[z]\langle w_3,w_4\rangle\big\}$ and the two blocking sets $U_1 = \{z\}$ and $U_2 = \{w_1,w_2\}$.

8.2 A lower bound assuming an extension of the LHC property

When we want to prove lower bounds on the blob-pebbling price of layered graphs, a useful generalization of Property 7.11 on page 513 turns out to be the following.

Property 8.7 (Generalized limited hiding-cardinality property)**.** We say that a blob-pebbling configuration $\mathbb{S}$ on a layered DAG G has the *generalized limited hiding-cardinality property with parameter K* if there is a vertex set U such that

1. U blocks $\mathbb{S}$,
2. $\mathrm{pot}(\mathbb{S}) = m(U)$, i. e., U is a minimum-measure blocker of $\mathbb{S}$,
3. $|U| \leq K \cdot \mathsf{cost}(\mathbb{S})$.

For brevity, in what follows we will just refer to this as the *generalized LHC property*.

We say that the graph G has the generalized LHC property with parameter K if all blob-pebbling configurations $\mathbb{S}$ on G have the generalized LHC property with parameter K.

When the parameter K is clear from context, we will just write that $\mathbb{S}$ or G has the generalized LHC property.

We claim that for all layered DAGs G_h of height h that have the generalized LHC property and are spreading, it holds that $\mathsf{Blob\text{-}Peb}(G_h) = \Theta(h)$. The proof of this fact is very much in the spirit of the proofs of Lemma 7.12 and Theorem 7.14, although the details are slightly more complicated.

Theorem 8.8 (Analogue of Theorem 7.14)**.** *Suppose that G_h is a layered DAG of height h possessing the generalized LHC property (Property 8.7) with some fixed parameter K. Then for any unconditional blob-pebbling $\mathcal{P} = \{\mathbb{S}_0 = \emptyset, \mathbb{S}_1, \ldots, \mathbb{S}_\tau\}$ of G_h it holds that*

$$\mathrm{pot}(\mathbb{S}_t) \leq (2K+1) \cdot \max_{s \leq t}\{\mathsf{cost}(\mathbb{S}_s)\}\,. \tag{8.2}$$

In particular, for any family of layered DAGs G_h that are also spreading in the sense of Definition 7.34, we have $\mathsf{Blob\text{-}Peb}(G_h) = \Theta(h)$.

We make two separate observations before presenting the proof.

Observation 8.9. *For any layered DAG G_h of height h it holds that $\mathsf{Blob\text{-}Peb}(G_h) = \mathrm{O}(h)$.*

Proof. Any layered DAG G_h can be black-pebbled with $h + \mathrm{O}(1)$ pebbles as noted in Theorem 7.2, and it is easy to see that a blob-pebbling can mimic a black pebbling in the same cost. ☐

Observation 8.10. *If G_h is a layered DAG of height h that is spreading in the sense of Definition 7.34, then $\mathrm{pot}_{G_h}([z]\langle\emptyset\rangle) = h+2$.*

Proof. The proof is fairly similar to the corresponding case for pyramids in Lemma 7.13. Note, though, that in contrast to Lemma 7.13, here we cannot get the statement from the generalized LHC property, but instead have to prove it directly.

Since $[z]$ is an atomic blob, the blocking and hiding relations coincide. The set $U = \{z\}$ hides itself and has measure $h+2$. We show that any other blocking set must have strictly larger measure.

Suppose that z is hidden by some vertex set $U' \neq \{z\}$. This U' is minimal without loss of generality. In particular, we can assume that U' is tight in the sense of Definition 7.15 and that $U' = U'\llbracket z \rrbracket$. Then by Corollary 7.23 it holds that U' is hiding-connected. Letting $L = \text{minlevel}\big(U'\big)$ and setting $j = h$ in the spreading inequality (7.10), we get that $\big|U'\big| \geq 1+h-L$ and hence $m\big(U'\big) \geq m^L\big(U'\big) \geq L+2(1+h-L) = 2h-L+2 > h+2$ since $L < h$. □

Proof of Theorem 8.8. The statement in the theorem follows from Observations 8.9 and 8.10 combined with the inequality (8.2), so just as for Theorem 7.14 the crux of the matter is the induction proof needed to get this inequality.

Suppose that U_t is such that it blocks $\mathbb{S}_t$ and $\text{pot}(\mathbb{S}_t) = m(U_t)$. By the inductive hypothesis, we have that $\text{pot}(\mathbb{S}_t) \leq (2K+1) \cdot \max_{s \leq t}\{\mathit{cost}(\mathbb{S}_s)\}$. We want to show for $\mathbb{S}_{t+1}$ that $\text{pot}(\mathbb{S}_{t+1}) \leq (2K+1) \cdot \max_{s \leq t+1}\{\mathit{cost}(\mathbb{S}_s)\}$. Clearly, this follows if we can prove that

$$\text{pot}(\mathbb{S}_{t+1}) \leq \max\{\text{pot}(\mathbb{S}_t), (2K+1) \cdot \mathit{cost}(\mathbb{S}_t)\}\,. \tag{8.3}$$

We also note that if U_t blocks $\mathbb{S}_{t+1}$ we are done, since if so $\text{pot}(\mathbb{S}_{t+1}) \leq m(U_t) = \text{pot}(\mathbb{S}_t)$.

We make a case analysis depending on the type of move in Definition 4.6 made to get from $\mathbb{S}_t$ to $\mathbb{S}_{t+1}$. Observe that a technical complication here is that multiple mergers, inflations and erasures can be joined into one single blob-pebbling move (potentially to avoid paying for expensive intermediate blob-pebbling configurations), so our proof has to work regardless of how pebbling steps are combined into moves.

This is not really a problem, however. We show below that if $\mathbb{S}' \rightsquigarrow \mathbb{S}$ is a pebbling step and U' blocks $\mathbb{S}'$, then U' also blocks $\mathbb{S}$ as long as the pebbling step is not an introduction (just as in the proof of Lemma 7.12). Therefore, the only case we have reason to worry about is when the pebbling makes an introduction move—in fact, an introduction move on a source vertex—and for such moves we will use the generalized LHC property to prove the bound that we need. That is, what we do is in effect to prove (8.2) but with the maximum on the right-hand side taken only over subconfigurations resulting from introduction moves. Clearly, this stronger inequality is sufficient to establish the theorem. The formal proof follows.

Erasure $\mathbb{S} = \mathbb{S}' \setminus \big\{[B]\langle W\rangle\big\}$ for $[B]\langle W\rangle \in \mathbb{S}'$. If U' blocks $\mathbb{S}'$, then obviously U' blocks $\mathbb{S} \subseteq \mathbb{S}'$.

Inflation $\mathbb{S} = \mathbb{S}' \cup \big\{[B]\langle W\rangle\big\}$ for $[B]\langle W\rangle$ inflated from some $[B']\langle W'\rangle \in \mathbb{S}'$ such that $B' \subseteq B$ and $W' \subseteq W$. Again we claim that if U' blocks $\mathbb{S}'$, then U' blocks $[B]\langle W\rangle$ and thus all of $\mathbb{S}$. To see this, suppose that P is any source path agreeing with B. Then this path also agrees with B', and so must be blocked by $U' \cup W'$ by assumption. Hence, it is certainly blocked by $U' \cup W \supseteq U' \cup W'$.

Merger $\mathbb{S} = \mathbb{S}' \cup \big\{[B]\langle W\rangle\big\}$ for $[B]\langle W\rangle$ derived by merger of $[B_1]\langle W_1\rangle, [B_2]\langle W_2\rangle \in \mathbb{S}'$ such that

$$B_1 \cap W_2 = \emptyset\,, \tag{8.4a}$$

$$B_2 \cap W_1 = \{v^*\}\,, \tag{8.4b}$$

$$B = (B_1 \cup B_2) \setminus \{v^*\}\,, \text{ and} \tag{8.4c}$$

$$W = (W_1 \cup W_2) \setminus \{v^*\}\,. \tag{8.4d}$$

Let us prove that if a set of vertices U' blocks two subconfigurations $[B_1]\langle W_1\rangle$ and $[B_2]\langle W_2\rangle$, then it must also block their merger. We have two cases. First, let P be any path via B, and suppose in addition that P visits the merger vertex v^*. If so, P agrees with B_2 and must be blocked by $U' \cup W_2$ by assumption. If on the other hand P agrees with B but does *not* visit v^*, then it is a path via B_1 that in addition does not pass through the white pebble v^* in W_1 eliminated in the merger. This means that $U' \cup (W_1 \setminus \{v^*\})$ must block P. In both cases we see that P is blocked by $U' \cup (W_1 \setminus \{v^*\}) \cup W_2 = U' \cup (W_1 \cup W_2) \setminus \{v^*\}$.

Introduction $\mathbb{S}_{t+1} = \mathbb{S}_t \cup \big\{[v]\langle pred(v)\rangle\big\}$. (Note that since introduction steps are always pebbling moves, we write $\mathbb{S}_t$ and $\mathbb{S}_{t+1}$ here instead of $\mathbb{S}'$ and $\mathbb{S}$.) Clearly, U_t blocks $\mathbb{S}_{t+1}$ if v is a non-source vertex, i. e., if $pred(v) \neq \emptyset$, since U_t blocks $\mathbb{S}_t$ and $[v]\langle pred(v)\rangle$ blocks itself. Hence, the only introduction moves we have to consider are for source vertices v, so that the subconfiguration introduced is $[v]\langle\emptyset\rangle$.

As in the proof of Lemma 7.12, U_t does not necessarily block $\mathbb{S}_{t+1}$ any longer but $U_{t+1} = U_t \cup \{v\}$ clearly does. For $j > 0$, it holds that $U_{t+1}\{\succeq j\} = U_t\{\succeq j\}$ and thus $m^j(U_{t+1}) = m^j(U_t)$. On the bottom level $j = 0$, using that $|U_t| \leq K \cdot \mathit{cost}(\mathbb{S}_t)$ by generalized LHC property (Property 8.7) we have

$$\begin{aligned} m^0(U_{t+1}) = 2 \cdot |U_{t+1}| = 2 \cdot (|U_t| + 1) \leq \\ 2 \cdot \big(K \cdot \mathit{cost}(\mathbb{S}_t) + 1\big) \leq 2 \cdot \big(K \cdot \mathit{cost}(\mathbb{S}_{t+1}) + 1\big) \leq \\ 2 \cdot \big(K \cdot \mathit{cost}(\mathbb{S}_{t+1}) + \mathit{cost}(\mathbb{S}_{t+1})\big) \leq 2(K+1) \cdot \mathit{cost}(\mathbb{S}_{t+1}) \end{aligned} \tag{8.5}$$

and we get that

$$\begin{aligned} \mathrm{pot}(\mathbb{S}_{t+1}) \leq m(U_{t+1}) \leq \max_j\big\{m^j(U_{t+1})\big\} \\ \leq \max\big\{m(U_t), (2K+1) \cdot \mathit{cost}(\mathbb{S}_{t+1})\big\} = \\ \max\big\{\mathrm{pot}(\mathbb{S}_t), (2K+1) \cdot \mathit{cost}(\mathbb{S}_{t+1})\big\} \end{aligned} \tag{8.6}$$

which is what is needed for the induction step to go through.

We see that regardless of the pebbling move(s) made in the transition $\mathbb{S}_t \rightsquigarrow \mathbb{S}_{t+1}$, the inequality (8.3) holds. The theorem follows by the induction principle. □

Now in order to prove a lower bound on $\mathit{Blob\text{-}Peb}(G_h)$ for layered spreading graphs G_h, all that remains is to find some constant K such that these DAGs can be shown to possess the generalized LHC property (Property 8.7) with parameter K. We do so in Sections 8.3 and 8.4 below.

8.3 Some structural transformations

As we tried to indicate by presenting the small toy blob-pebbling configurations in Examples 8.4 and 8.5, the potential in the blob-pebble game behaves somewhat differently from the potential in the standard pebble game. There are (at least) two important differences:

- Firstly, for the white pebbles we have to keep track of exactly which black pebbles they can help to block. This can lead to slightly unexpected consequences such as the blocking set U and the set of white pebbles overlapping.
- Secondly, for black blobs there is a much wider choice where to block the blob-pebbles than for atomic pebbles. It seems that to minimize the potential, blocking black blobs on (reasonably) low levels should still be a good idea. However, we cannot a priori exclude the possibility that if a lot of black blobs intersect in some high-level vertex, adding this vertex to a blocking set U might be a better idea.

In this subsection we address the first of these issues. The second issue, which turns out to be much trickier, is dealt with in the next subsection.

One simplifying observation is that we do not have to prove Property 8.7 for arbitrary blob-pebbling configurations. Below, we show that one can do some technical preprocessing of the blob-pebbling configurations so that it suffices to prove the generalized LHC property for the subclass of configurations resulting from this preprocessing.[6] Throughout this subsection, we assume that the parameter K is some fixed constant.

We start gently by taking care of a pretty obvious redundancy. Let us say that the blob subconfiguration $[B]\langle W\rangle$ is *self-blocking* if W blocks B. The blob-pebbling configuration $\mathbb{S}$ is *self-blocker-free* if there are no self-blocking subconfigurations in $\mathbb{S}$. That is, if $[B]\langle W\rangle$ is self-blocking, W needs no extra help blocking B. Perhaps the simplest example of this is $[B]\langle W\rangle = [v]\langle \mathit{pred}(v)\rangle$ for a non-source vertex v. Another example of a self-blocking subconfiguration $[B]\langle W\rangle$ is when B is not a chain, as discussed after Definition 8.1. The following proposition is immediate.

Proposition 8.11. *For $\mathbb{S}$ any blob-pebbling configuration, let $\mathbb{S}'$ be the blob-pebbling configuration with all self-blockers in $\mathbb{S}$ removed. Then $\mathit{cost}(\mathbb{S}') \leq \mathit{cost}(\mathbb{S})$, $\mathrm{pot}(\mathbb{S}') = \mathrm{pot}(\mathbb{S})$ and any blocking set U' for $\mathbb{S}'$ is also a blocking set for $\mathbb{S}$.*

Corollary 8.12. *Suppose that the generalized LHC property holds for self-blocker-free blob-pebbling configurations. Then the generalized LHC property holds for all blob-pebbling configurations.*

Proof. If $\mathbb{S}$ is *not* self-blocker-free, take the maximal $\mathbb{S}' \subseteq \mathbb{S}$ that is and the blocking set U' that the generalized LHC property provides for this $\mathbb{S}'$. Then U' blocks $\mathbb{S}$ and since the two configurations $\mathbb{S}$ and $\mathbb{S}'$ have the same blocking sets their potentials are equal, so $\mathrm{pot}(\mathbb{S}) = m(U')$. Finally, we have that $|U| \leq K \cdot \mathit{cost}(\mathbb{S}') \leq K \cdot \mathit{cost}(\mathbb{S})$. Thus the generalized LHC property holds for $\mathbb{S}$. ☐

Observe that this means that in what follows, we can without loss of generality assume that all blob-pebbles B are chains. In particular, we can assume (by mild abuse of notation) that $\mathrm{bot}(B)$ is not a set but a single well-defined vertex.

We now move on to a more interesting observation. Looking at $\mathbb{S} = \{[z]\langle y_1\rangle, [z]\langle y_2\rangle\}$ in Example 8.4, it seems that the white pebbles really do not help at all. One might ask if we could not just throw them

[6] Note that we did something similar in Section 7.3 after Lemma 7.18, when we argued that if U is a minimum-measure hiding set for $\mathbb{P} = (B,W)$, we can assume without loss of generality that $U \cup W$ is tight. For if not, we just prove the limited hiding-cardinality property for some tight subset $U' \cup W' \subseteq U \cup W$ instead. This is completely analogous to the reasoning here, but since matters become more complex we need to be a bit more careful.

away? Perhaps somewhat surprisingly, the answer is yes, and we can capture the intuitive concept of necessary white pebbles and formalize it as follows in the definitions below.

Definition 8.13 (White sharpening)**.** Given $\mathbb{S} = \left\{[B_i]\langle W_i\rangle\right\}_{i\in[m]}$, we say that $\mathbb{S}'$ is a *white sharpening* of $\mathbb{S}$ if $\mathbb{S}' = \left\{[B'_i]\langle W'_i\rangle\right\}_{i\in[m]}$ for $B'_i = B_i$ and $W'_i \subseteq W_i$.

That is, a white sharpening removes white pebbles and thus makes the blob-pebbling configuration stronger or "sharper" in the sense that the cost can only decrease and the potential can only increase.

Proposition 8.14. *If $\mathbb{S}'$ is a white sharpening of $\mathbb{S}$ it holds that $\mathit{cost}(\mathbb{S}') \leq \mathit{cost}(\mathbb{S})$ and $\mathrm{pot}(\mathbb{S}') \geq \mathrm{pot}(\mathbb{S})$. More precisely, any blocking set U' for $\mathbb{S}'$ is also a blocking set for $\mathbb{S}$.*

Proof. The statement about cost is immediate from Definition 4.7. The statement about potential clearly follows from Definition 8.3 since it holds that any blocking set U' for $\mathbb{S}'$ is also a blocking set for $\mathbb{S}$. $\square$

In the next definition, we suppose that there is some fixed but arbitrary ordering of the vertices in G, and that the vertices are considered in this order.

Definition 8.15 (White elimination)**.** For $[B]\langle W\rangle$ a blob subconfiguration and U any blocking set for $[B]\langle W\rangle$, write $W = \{w_1,\ldots,w_s\}$, set $W^0 := W$ and iteratively perform the following for $i = 1,\ldots,s$: If $U \cup (W^{i-1} \setminus \{w_i\})$ blocks B, set $W^i := W^{i-1} \setminus \{w_i\}$, otherwise set $W^i := W^{i-1}$. We define the *white elimination* of $[B]\langle W\rangle$ with respect to U to be $\mathcal{W}\text{-elim}([B]\langle W\rangle, U) = [B]\langle W^s\rangle$ for W^s the final set resulting from the procedure above.

For $\mathbb{S}$ a blob-pebbling configuration and U a blocking set for $\mathbb{S}$, we define

$$\mathcal{W}\text{-elim}(\mathbb{S},U) = \left\{\mathcal{W}\text{-elim}([B]\langle W\rangle,U)\,\middle|\,[B]\langle W\rangle \in \mathbb{S}\right\}. \tag{8.7}$$

We say that the elimination is *strict* if $\mathbb{S} \neq \mathcal{W}\text{-elim}(\mathbb{S},U)$. If $\mathbb{S} = \mathcal{W}\text{-elim}(\mathbb{S},U)$ we say that $\mathbb{S}$ is *white-eliminated*, or *$\mathcal{W}$-eliminated* for short, with respect to U.

Clearly $\mathcal{W}\text{-elim}(\mathbb{S},U)$ is a white sharpening of $\mathbb{S}$. And if we pick the right U, we simplify the problem of proving the generalized LHC property a bit more.

Lemma 8.16. *If U is a minimum-measure blocking set for $\mathbb{S}$, then $\mathbb{S}' = \mathcal{W}\text{-elim}(\mathbb{S},U)$ is a white sharpening of $\mathbb{S}$ such that $\mathrm{pot}(\mathbb{S}') = \mathrm{pot}(\mathbb{S})$ and U blocks $\mathbb{S}'$.*

Proof. Since $\mathbb{S}' = \mathcal{W}\text{-elim}(\mathbb{S},U)$ is a white sharpening of $\mathbb{S}$ (which is easily verified from Definitions 8.13 and 8.15), it holds by Proposition 8.14 that $\mathrm{pot}(\mathbb{S}') \geq \mathrm{pot}(\mathbb{S})$. Looking at the construction in Definition 8.15, we also see that the white pebbles are "sharpened away" with care so that U remains a blocking set. Thus $m(U) \geq \mathrm{pot}(\mathbb{S}') = \mathrm{pot}(\mathbb{S}) = m(U)$, and the lemma follows. $\square$

Corollary 8.17. *Suppose that the generalized LHC property holds for the set of all blob-pebbling configurations $\mathbb{S}$ having the property that for all minimum-measure blocking sets U for $\mathbb{S}$ it holds that $\mathbb{S} = \mathcal{W}\text{-elim}(\mathbb{S},U)$. Then the generalized LHC property holds for all blob-pebbling configurations.*

Proof. This is essentially the same reasoning as in the proof of Corollary 8.12 plus induction. Let $\mathbb{S}$ be any blob-pebbling configuration. Suppose that there exists a minimum-measure blocker U for $\mathbb{S}$ such that $\mathbb{S}$ is not $\mathcal{W}$-eliminated with respect to U. Let $\mathbb{S}^1 = \mathcal{W}\text{-elim}(\mathbb{S}, U)$. Then $\mathit{cost}(\mathbb{S}^1) \leq \mathit{cost}(\mathbb{S})$ by Proposition 8.14 and $\mathrm{pot}(\mathbb{S}^1) = \mathrm{pot}(\mathbb{S})$ by Lemma 8.16.

If there is a minimum-measure blocker U^1 for $\mathbb{S}^1$ such that $\mathbb{S}^1$ is not $\mathcal{W}$-eliminated with respect to U^1, set $\mathbb{S}^2 = \mathcal{W}\text{-elim}(\mathbb{S}^1, U^1)$. Continuing in this manner, we get a chain $\mathbb{S}^1, \mathbb{S}^2, \mathbb{S}^3, \ldots$ of strict $\mathcal{W}$-eliminations such that $\mathit{cost}(\mathbb{S}^1) \geq \mathit{cost}(\mathbb{S}^2) \geq \mathit{cost}(\mathbb{S}^3) \ldots$ and $\mathrm{pot}(\mathbb{S}^1) = \mathrm{pot}(\mathbb{S}^2) = \mathrm{pot}(\mathbb{S}^3) = \ldots$ This chain must terminate at some configuration $\mathbb{S}^k$ since the total number of white pebbles (counted with repetitions) decreases in every round.

Let U^k be the blocker that the generalized LHC property provides for $\mathbb{S}^k$. Then U^k blocks $\mathbb{S}$, $\mathrm{pot}(\mathbb{S}) = \mathrm{pot}(\mathbb{S}^k) = m(U^k)$, and $|U^k| \leq K \cdot \mathit{cost}(\mathbb{S}^k) \leq K \cdot \mathit{cost}(\mathbb{S})$. Thus the generalized LHC property holds for $\mathbb{S}$. $\square$

We note that in particular, it follows from the construction in Definition 8.15 combined with Corollary 8.17 that we can assume without loss of generality for any blocking set U and any blob-pebbling configuration $\mathbb{S}$ that U does not intersect the set of white-pebbled vertices in $\mathbb{S}$.

Proposition 8.18. *If* $\mathbb{S} = \mathcal{W}\text{-elim}(\mathbb{S}, U)$*, then in particular it holds that* $U \cap \mathcal{W}(\mathbb{S}) = \emptyset$*.*

Proof. Any $w \in \mathcal{W}(\mathbb{S}) \cap U$ would have been removed in the $\mathcal{W}$-elimination. $\square$

8.4 Proof of the generalized limited hiding-cardinality property

We are now ready to embark on the proof of the generalized LHC property for layered spreading DAGs.

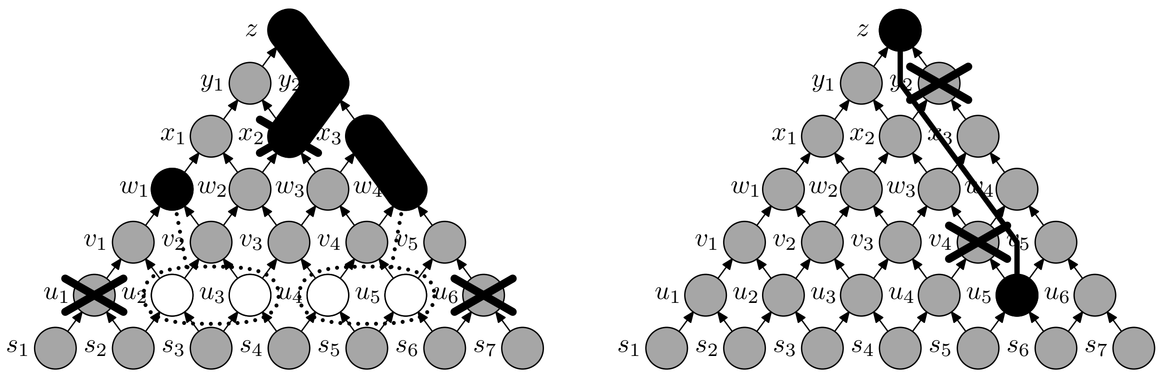
Theorem 8.19. *All layered DAGs that are spreading possess the generalized limited hiding-cardinality property (Property 8.7) with parameter* $K = 13$*.*

Since pyramids are spreading graphs by Theorem 7.35, this is all that we need to get the lower bound on blob-pebbling price on pyramids from Theorem 8.8. We remark that the parameter K in Theorem 8.19 can easily be improved with a little extra work. However, our main concern here is not optimality of constants but clarity of exposition.

We prove Theorem 8.19 by applying the preprocessing in the previous subsection and then (almost) reducing the problem to the standard black-white pebble game. However, some twists are added along the way since our potential measure for blobs behave differently from Klawe's potential measure for black and white pebbles. Let us first exemplify two problems that arise if we try to naively mimic step by step Klawe's proof for the standard black-white pebble game.

The first problem is that in the standard black-white pebble game, if U is a minimum-measure hiding set for $\mathbb{P} = (B, W)$, Lemma 7.18 tells us that we can assume without loss of generality that $U \cup W$ is tight. This is *not* true in the blob-pebble game, not even after the transformations in Section 8.3.

Example 8.20. Consider the configuration $\mathbb{S} = \{[w_1]\langle u_2, u_3\rangle, [w_4, x_3]\langle u_4, u_5\rangle, [x_2, y_2, z]\langle\emptyset\rangle\}$ with blocking set $U = \{x_2, u_1, u_6\}$ in Figure 12(a). It can be verified that U is a minimum-measure blocking set and that the configuration $\mathbb{S}$ is $\mathcal{W}$-eliminated with respect to U, but the set $U \cup \mathcal{W}(\mathbb{S}) = \{u_1, u_2, u_3, u_4, u_5, u_6, x_2\}$ is not tight (because of x_2).



(a) Minimum-measure but non-tight blocking set.

(b) Tight but non-connected blocker for blob.

Figure 12: Two blob-pebbling configurations with problematic blocking sets.

This can be handled, but a more serious second problem is that even if the set $U \cup W$ blocking the chain B is tight, there is no guarantee that the vertices in $U \cup W$ end up in the same connected component of the hiding set graph $\mathcal{H}(U \cup W)$ in Definition 7.20.

Example 8.21. Consider the single-blob configuration $\mathbb{S} = \{[u_5, z]\langle\emptyset\rangle\}$ in Figure 12(b). It is easy to verify that $U = \{v_4, y_2\}$ is a subset-minimal blocker of $\mathbb{S}$ and also a tight vertex set. This highlights the fact that blocking sets for blob-pebbling configurations can have rather different properties than hiding sets for standard pebbles. In particular, a minimal blocking set for a single blob can have several "isolated" vertices at large distances from one another. Among other problems, this leads to difficulties in defining connected components of blocking sets for subconfigurations.

The naive attempt to generalize Definition 7.20 of connected components in a hiding set graph to blocking sets would place the vertices v_4 and y_2 in different connected components $\{v_4\}$ and $\{y_2\}$, none of which blocks $\mathbb{S} = \{[u_5, z]\langle\emptyset\rangle\}$. This is not what we want (compare Corollary 7.23 for hiding sets for black-white pebble configurations). We remark that there really cannot be any other sensible definition that places v_4 and y_2 in the same connected component either, at least not if we want to appeal to the spreading properties in Definition 7.34. Since the level difference in U is 3 but the size of the set is only 2, the spreading inequality (7.10) cannot hold for this set.

To get around this problem, we will instead use connected components defined in terms of hiding the singleton black pebbles given by the bottom vertices of our blobs. For a start, recalling Definitions 7.6 and 8.1, let us make an easy observation relating the hiding and blocking relations for a blob.

Observation 8.22. *If a vertex set V hides some vertex $b \in B$, then V blocks B.*

Proof. If V blocks all paths visiting b, then in particular it blocks the subset of paths that not only visits b but agree with all of B. $\square$

We will focus on the case when the unique bottom vertex of a chain blob is hidden.

Definition 8.23 (Hiding blob-pebbling configurations)**.** We say that the vertex set U *hides* the subconfiguration $[B]\langle W\rangle$ if B is a chain and $U \cup W$ hides the vertex $\mathrm{bot}(B)$. U is said to hide the blob-pebbling configuration $\mathbb{S}$ if U hides all $[B]\langle W\rangle \in \mathbb{S}$.

If U does not hide $[B]\langle W\rangle$, then U blocks $[B]\langle W\rangle$ only if $U \cap G^{\triangledown}_{\mathrm{bot}(B)}$ does.

Proposition 8.24. *Suppose that a vertex set U in a layered DAG G blocks but does not hide the subconfiguration $[B]\langle W\rangle$ and that $[B]\langle W\rangle$ does not block itself. Then $U \cap G^{\mathrm{bot}(B)}_{\vartriangle}$ does not block $[B]\langle W\rangle$, but there is a subset $U' \subseteq U \cap G^{\triangledown}_{\mathrm{bot}(B)}$ that blocks $[B]\langle W\rangle$.*

Proof. Suppose that $U \cup W$ blocks B but does not hide $b = \mathrm{bot}(B)$, and that W does not block B. Then there is a source path P_2 via B such that $P_2 \cap W = \emptyset$. Also, there is a source path P_1 to b such that $P_1 \cap (U \cup W) = \emptyset$. Let $P = \left(P_1 \cap G^{b}_{\vartriangle}\right) \cup \left(P_2 \cap G^{\triangledown}_{b}\right)$ be the source path that starts like P_1 and continues like P_2 from b onwards. Clearly,

$$P \cap \left(\left(U \cap G^{b}_{\vartriangle}\right) \cup W\right) = \left(P_1 \cap (U \cup W)\right) \cup \left(P_2 \cap W\right) = \emptyset \tag{8.8}$$

so $U \cap G^{b}_{\vartriangle}$ does not block $[B]\langle W\rangle$.

Suppose that $U \cap G^{\triangledown}_{b}$ does not block $[B]\langle W\rangle$. Since $U \cup W$ does not hide b, there is some source path P_1 to b with $P_1 \cap (U \cup W) = \emptyset$. Also, since $U \cup W$ blocks B but $\left(U \cap G^{\triangledown}_{b}\right) \cup W$ does not, there is a source path P_2 via B such that $P_2 \cap (U \cup W) \neq \emptyset$ but $P_2 \cap (U \cup W) \cap G^{\triangledown}_{b} = \emptyset$. But then let $P = \left(P_1 \cap G^{b}_{\vartriangle}\right) \cup \left(P_2 \cap G^{\triangledown}_{b}\right)$ be the source path that starts like P_1 and continues like P_2 from b onwards. We get that P agrees with B and that $P \cap (U \cup W) = \emptyset$, contradicting the assumption that U blocks $[B]\langle W\rangle$. $\square$

We want to distinguish between subconfigurations that are hidden and subconfigurations that are just blocked, but not hidden. To this end, let us introduce the notation

$$\mathbb{S}_H(\mathbb{S}, U) = \left\{[B]\langle W\rangle \in \mathbb{S} \,\middle|\, U \text{ hides } [B]\langle W\rangle\right\} \tag{8.9}$$

to denote the subconfigurations in $\mathbb{S}$ hidden by U and

$$\mathbb{S}_B(\mathbb{S}, U) = \mathbb{S} \setminus \mathbb{S}_H(\mathbb{S}, U) \tag{8.10}$$

to denote the subconfigurations that are just blocked. We write

$$\mathcal{B}_H(\mathbb{S}, U) = \{\mathrm{bot}(B) \mid [B]\langle W\rangle \in \mathbb{S}_H(\mathbb{S}, U)\} \tag{8.11}$$

and

$$\mathcal{B}_B(\mathbb{S}, U) = \{\mathrm{bot}(B) \mid [B]\langle W\rangle \in \mathbb{S}_B(\mathbb{S}, U)\} \tag{8.12}$$

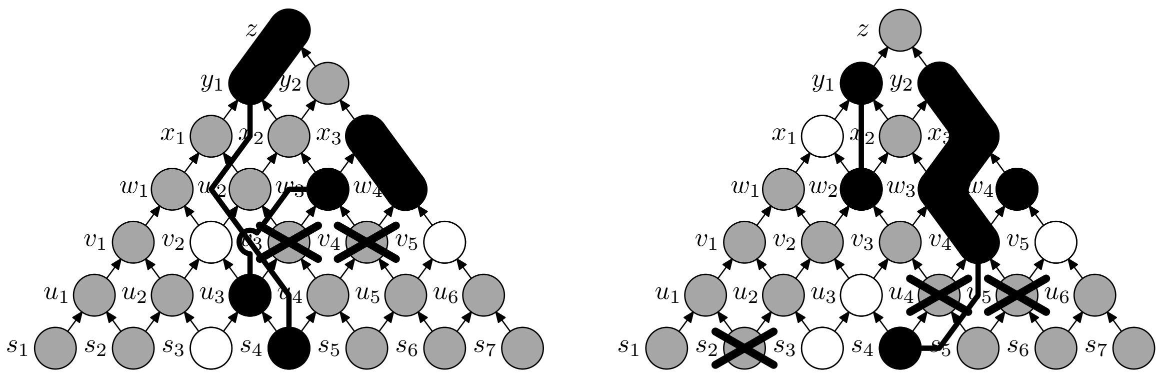
to denote the black bottom vertices in these two subsets of subconfigurations and note that we can have $\mathcal{B}_H(\mathbb{S}, U) \cap \mathcal{B}_B(\mathbb{S}, U) \neq \emptyset$. The white pebbles in these subsets located below the bottom vertices of the black blobs that they are supporting are denoted

$$\mathcal{W}^{\vartriangle}_H(\mathbb{S}, U) = \left\{W \cap G^{b}_{\vartriangle} \,\middle|\, [B]\langle W\rangle \in \mathbb{S}_H(\mathbb{S}, U), b = \mathrm{bot}(B)\right\} \tag{8.13}$$

and

$$\mathcal{W}^{\vartriangle}_B(\mathbb{S}, U) = \left\{W \cap G^{b}_{\vartriangle} \,\middle|\, [B]\langle W\rangle \in \mathbb{S}_B(\mathbb{S}, U), b = \mathrm{bot}(B)\right\}. \tag{8.14}$$

This notation will be used heavily in what follows, so we give a couple of simple but hopefully illuminating examples before we continue.



(a) $\{[s_4,y_1,z]\langle v_2\rangle,[u_3,w_3]\langle s_3\rangle,[w_4,x_3]\langle v_5\rangle\}$.

(b) $\{[s_4,v_4,w_3,x_3,y_2]\langle\emptyset\rangle,[w_2,y_1]\langle s_3,u_3,x_1\rangle,[w_4]\langle v_5\rangle\}$.

Figure 13: Examples of blob-pebbling configurations with hidden and just blocked blobs.

Example 8.25. Consider the blob-pebbling configurations and blocking sets in Figure 13. For the blob-pebbling configuration $\mathbb{S}_1 = \{[s_4,y_1,z]\langle v_2\rangle,[u_3,w_3]\langle s_3\rangle,[w_4,x_3]\langle v_5\rangle\}$ with blocking set $U_1 = \{v_3,v_4\}$ in Figure 13(a), the vertex set $\{v_4,v_5\}$ hides $w_4 = \mathrm{bot}([w_4,x_3])$ but $[s_4,y_1,z]$ is blocked but not hidden by $\{v_2,v_3,v_4\}$ and $[u_3,w_3]$ is blocked but not hidden by $\{v_3\}$. Thus, we have

$$
\begin{aligned}
\mathbb{S}_H(\mathbb{S}_1,U_1) &= \{[w_4,x_3]\langle v_5\rangle\}\\
\mathbb{S}_B(\mathbb{S}_1,U_1) &= \{[s_4,y_1,z]\langle v_2\rangle,[u_3,w_3]\langle s_3\rangle\}\\
\mathcal{B}_H(\mathbb{S}_1,U_1) &= \{w_4\}\\
\mathcal{B}_B(\mathbb{S}_1,U_1) &= \{s_4,u_3\}\\
\mathcal{W}^{\triangle}_H(\mathbb{S}_1,U_1) &= \{v_5\}\\
\mathcal{W}^{\triangle}_B(\mathbb{S}_1,U_1) &= \{s_3\}
\end{aligned}
$$

in this example. For the configuration $\mathbb{S}_2 = \{[s_4,v_4,w_3,x_3,y_2]\langle\emptyset\rangle,[w_2,y_1]\langle s_3,u_3,x_1\rangle,[w_4]\langle v_5\rangle\}$ with blocker $U_2 = \{s_2,u_4,u_5\}$ in Figure 13(b), it is straightforward to verify that

$$
\begin{aligned}
\mathbb{S}_H(\mathbb{S}_2,U_2) &= \{[w_2,y_1]\langle s_3,u_3,x_1\rangle,[w_4]\langle v_5\rangle\}\\
\mathbb{S}_B(\mathbb{S}_2,U_2) &= \{[s_4,v_4,w_3,x_3,y_2]\langle\emptyset\rangle\}\\
\mathcal{B}_H(\mathbb{S}_2,U_2) &= \{w_2,w_4\}\\
\mathcal{B}_B(\mathbb{S}_2,U_2) &= \{s_4\}\\
\mathcal{W}^{\triangle}_H(\mathbb{S}_2,U_2) &= \{s_3,u_3,v_5\}\\
\mathcal{W}^{\triangle}_B(\mathbb{S}_2,U_2) &= \emptyset
\end{aligned}
$$

are the corresponding sets.

Let us also use the opportunity to illustrate Definition 8.15. The blob-pebbling configuration $\mathbb{S}_1$ is not $\mathcal{W}$-eliminated with respect to U_1, since U_1 also blocks this configuration with the white pebble on s_3 removed. However, a better idea measure-wise is to change the blocking set for $\mathbb{S}_1$ to $U_1' = \{s_4,v_4\}$, which has measure $m(U_1') = 4 < 6 = m(U_1)$. The vertex set U_2 can be verified to be a minimum-measure blocker for $\mathbb{S}_2$, but when $\mathbb{S}_2$ is $\mathcal{W}$-eliminated with respect to U_2 the white pebble on x_1 disappears.

For the rest of this section we will assume without loss of generality (in view of Proposition 8.11 and Corollary 8.17) that we are dealing with a blob-pebbling configuration $\mathbb{S}$ and a minimum-measure blocker U of $\mathbb{S}$ such that $\mathbb{S}$ is free from self-blocking subconfigurations and is $\mathcal{W}$-eliminated with respect to U. As an aside, we note that it is not hard to show (using Definition 8.15 and Proposition 8.24) that this implies that $\mathcal{W}_B^{\triangle}(\mathbb{S}, U) = \emptyset$. We will tend to drop the arguments $\mathbb{S}$ and U for $\mathbb{S}_H, \mathbb{S}_B, \mathcal{B}_H, \mathcal{B}_B, \mathcal{W}_H^{\triangle}$, and $\mathcal{W}_B^{\triangle}$, since from now on the blob-pebbling configuration $\mathbb{S}$ and the blocker U will be fixed. With this notation, Theorem 8.19 clearly follows if we can prove the following lemma.

Lemma 8.26. *Let $\mathbb{S}$ be any blob-pebbling configuration on a layered spreading DAG and U be any blocking set for $\mathbb{S}$ such that*

1. $\operatorname{pot}(\mathbb{S}) = m(U)$*, i. e., U is a minimum-measure blocker of $\mathbb{S}$,*
2. *$\mathbb{S}$ is free from self-blocking subconfigurations and is $\mathcal{W}$-eliminated with respect to U, and*
3. *U has minimal size among all blocking sets U' for $\mathbb{S}$ such that* $\operatorname{pot}(\mathbb{S}) = m(U')$.

Then $|U| \le 13 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|$.

The proof is by contradiction, although we will have to work harder than for the corresponding Theorem 7.25 for black-white pebbling and also use (the proof of) the latter theorem as a subroutine. Thus, for the rest of this section, let us assume on the contrary that U has all the properties stated in Lemma 8.26 but that $|U| > 13 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|$. We will show that this leads to a contradiction.

For the subconfiguration in $\mathbb{S}_H$ that are hidden by U, one could argue that matters should be reasonably similar to the case for standard black-white pebbling, and hopefully we could apply similar reasoning as in Section 7.3 to prove something useful about the vertex set hiding these subconfigurations. The subconfigurations in $\mathbb{S}_B$ that are just blocked but not hidden, however, seem harder to get a handle on (compare Example 8.21).

Let $U_H \subseteq U$ be a smallest vertex set hiding $\mathbb{S}_H$ and let $U_B = U \setminus U_H$. The set U_B consists of vertices that are not involved in any hiding of subconfigurations in $\mathbb{S}_H$, but only in blocking subconfigurations in $\mathbb{S}_B$ on levels above their bottom vertices. As a first step towards proving Lemma 8.26, and thus Theorem 8.19, we want to argue that U_B cannot be very large.

Consider the blobs in $\mathbb{S}_B$. By definition they are not hidden, but are blocked at some position above $\operatorname{level}(\operatorname{bot}(B))$. Since the vertices in U_B are located on high levels, a naive attempt to improve the blocking set would be to pick some $u \in U_B$ and replace it by the vertices in $\mathcal{B}_B$ corresponding to the subconfigurations in $\mathbb{S}_B$ that u is involved in blocking, i. e., by the set

$$\mathcal{B}^u = \left\{\operatorname{bot}(B) \middle| U \setminus \{u\} \text{ does not block } [B]\langle W\rangle \in \mathbb{S}_B\right\}. \tag{8.15}$$

Note that $\mathcal{B}^u$ is lower down in the graph than u, so $(U \setminus \{u\}) \cup \mathcal{B}^u$ is obtained from U by moving vertices downwards and by construction $(U \setminus \{u\}) \cup \mathcal{B}^u$ blocks $\mathbb{S}$. But by assumption, U has minimal potential and cardinality, so this new blocking set cannot be an improvement measure- or cardinality-wise. The same holds if we extend the construction to subsets $U' \subseteq U_B$ and the corresponding bottom vertices

$\mathcal{B}^{U'} \subseteq \mathcal{B}_B$. By assumption we can never find any subset such that $(U \setminus \{U'\}) \cup \mathcal{B}^{U'}$ is a better blocker than U. It follows that the cost of the blobs that U_B helps to block must be larger than the size of U_B, and in particular that $|U_B| \leq |\mathcal{B}_B|$. Let us write this down as a lemma and prove it properly.

Lemma 8.27. *Let $\mathbb{S}$ be any blob-pebbling configuration on a layered DAG and U be any blocking set for $\mathbb{S}$ such that* $\mathrm{pot}(\mathbb{S}) = m(U)$*, U has minimal size among all blocking sets U' for $\mathbb{S}$ with* $\mathrm{pot}(\mathbb{S}) = m(U')$*, and $\mathbb{S}$ is free from self-blocking subconfigurations and is $\mathcal{W}$-eliminated with respect to U. Then if $U_H \subseteq U$ is any smallest set hiding $\mathbb{S}_H$ and $U_B = U \setminus U_H$, it holds that $|U_B| \leq |\mathcal{B}_B|$.*

Before proving this lemma, we note the immediate corollary that if the whole blocking set U is significantly larger than $cost(\mathbb{S})$, the lion's share of U by necessity consists not of vertices blocking subconfigurations in $\mathbb{S}_B$, but of vertices hiding subconfigurations in $\mathbb{S}_H$. And recall that we are indeed assuming, to get a contradiction, that U is large.

Corollary 8.28. *Assume that $\mathbb{S}$ and U are as in Lemma 8.26 but with $|U| > 13 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|$. Let $U_H \subseteq U$ be a smallest set hiding $\mathbb{S}_H$. Then $|U_H| > 12 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|$.*

As was indicated in the informal discussion preceding Lemma 8.27, the proof of the lemma uses the easy observation that moving vertices downwards can only decrease the measure.

Observation 8.29. *Suppose that U, V_1 and V_2 are vertex sets in a layered DAG such that $U \cap V_2 = \emptyset$ and there is a one-to-one (but not necessarily onto) mapping $f : V_1 \to V_2$ with the property that* $\mathrm{level}(v) \leq \mathrm{level}(f(v))$*. Then $m(U \cup V_1) \leq m(U \cup V_2)$.*

Proof. This follows immediately from Definition 7.8 on page 512 since the mapping f tells us that

$$\begin{aligned} |(U \cup V_1)\{\succeq j\}| \leq |U\{\succeq j\}| + |V_1\{\succeq j\}| &\leq |U\{\succeq j\}| + |f(V_1\{\succeq j\})| \\ &\leq |U\{\succeq j\}| + |V_2\{\succeq j\}| \leq |(U \cup V_2)\{\succeq j\}| \end{aligned} \tag{8.16}$$

for all j. □

Proof of Lemma 8.27. Note first that by Proposition 8.24, for every $[B]\langle W\rangle \in \mathbb{S}_B$ with $b = \mathrm{bot}(B)$ it holds that $U \cap G_b^{\triangledown} = (U_H \,\dot\cup\, U_B) \cap G_b^{\triangledown}$ blocks $[B]\langle W\rangle$. Therefore, all vertices in U_B needed to block $[B]\langle W\rangle$ can be found in $U_B \cap G_b^{\triangledown}$. Rephrasing this slightly, the blob-pebbling configuration $\mathbb{S}$ is blocked by $U_H \,\dot\cup\, \left(U_B \cap \bigcup_{b \in \mathcal{B}_B} G_b^{\triangledown}\right)$, and since U is subset-minimal we get that

$$U_B = U_B \cap \textstyle\bigcup_{b \in \mathcal{B}_B} G_b^{\triangledown} \,. \tag{8.17}$$

Consider the bipartite graph with $\mathcal{B}_B$ and U_B as the left- and right-hand vertices, where the neighbours of each $b \in \mathcal{B}_B$ are the vertices $N(b) = U_B \cap G_b^{\triangledown}$ in U_B above b. We have that $N(\mathcal{B}_B) = U_B \cap \bigcup_{b \in \mathcal{B}_B} G_b^{\triangledown} = U_B$ by (8.17). Let $\mathcal{B}' \subseteq \mathcal{B}_B$ be a largest set such that $\left|N(\mathcal{B}')\right| < \left|\mathcal{B}'\right|$. If $\mathcal{B}' = \mathcal{B}_B$ we are done since this is the inequality $|U_B| < |\mathcal{B}_B|$. Suppose therefore that $\mathcal{B}' \subsetneq \mathcal{B}_B$ and $|U_B| = |N(\mathcal{B}_B)| > |\mathcal{B}_B|$.

For all $\mathcal{B}'' \subseteq \mathcal{B}_B \setminus \mathcal{B}'$ we must have $\left|N(\mathcal{B}'') \setminus N(\mathcal{B}')\right| \geq \left|\mathcal{B}''\right|$, for otherwise $\mathcal{B}''$ could be added to $\mathcal{B}'$ to yield an even larger set $\mathcal{B}^* = \mathcal{B}' \cup \mathcal{B}''$ with $\left|N(\mathcal{B}^*)\right| < |\mathcal{B}^*|$ contrary to the assumption that $\mathcal{B}'$ has maximal size among all sets with this property. It follows by Hall's marriage theorem that there must exist a matching of $\mathcal{B}_B \setminus \mathcal{B}'$ into $N(\mathcal{B}_B \setminus \mathcal{B}') \setminus N(\mathcal{B}') = U_B \setminus N(\mathcal{B}')$. Thus, $\left|\mathcal{B}_B \setminus \mathcal{B}'\right| \leq \left|U_B \setminus N(\mathcal{B}')\right|$ and in addition it follows from the way our bipartite graph is constructed that every $b \in \mathcal{B}_B \setminus \mathcal{B}'$ is matched to some $u \in U_B \setminus N(\mathcal{B}')$ with $\text{level}(u) \geq \text{level}(b)$.

Clearly, all subconfigurations in

$$\mathbb{S}_B^1 = \left\{ [B]\langle W \rangle \in \mathbb{S}_B \,\middle|\, \text{bot}(B) \in \mathcal{B}_B \setminus \mathcal{B}' \right\} \tag{8.18}$$

are blocked by $\mathcal{B}_B \setminus \mathcal{B}'$ (even hidden by this set, to be precise). Also, as was argued in the beginning of the proof, every $[B]\langle W \rangle \in \mathbb{S}_B$ with $b = \text{bot}(B)$ is blocked by $U_H \cup \left(U_B \cap G_b^{\triangledown} \right) = U_H \cup N(b)$, so all subconfigurations in

$$\mathbb{S}_B^2 = \left\{ [B]\langle W \rangle \in \mathbb{S}_B \,\middle|\, \text{bot}(B) \in \mathcal{B}' \right\} \tag{8.19}$$

are blocked by $U_H \cup N(\mathcal{B}')$ where $\left| N(\mathcal{B}') \right| < \left| \mathcal{B}' \right|$. And we know that $\mathbb{S}_H$ is blocked (even hidden) by U_H. It follows that if we let

$$U^* = U_H \cup N(\mathcal{B}') \cup \left(\mathcal{B}_B \setminus \mathcal{B}' \right) \tag{8.20}$$

we get a vertex set U^* that blocks $\mathbb{S}_H \cup \mathbb{S}_B^1 \cup \mathbb{S}_B^2 = \mathbb{S}$, has measure $m(U^*) \leq m(U)$ because of Observation 8.29, and has size

$$\left| U^* \right| \leq |U_H| + \left| N(\mathcal{B}') \right| + \left| \mathcal{B}_B \setminus \mathcal{B}' \right| < |U_H| + \left| \mathcal{B}' \right| + \left| \mathcal{B}_B \setminus \mathcal{B}' \right| = |U| \tag{8.21}$$

strictly less than the size of U. But this is a contradiction, since U was chosen to be of minimal size. The lemma follows. □

The idea in the remaining part of the proof is as follows: Fix some smallest subset $U_H \subseteq U$ that hides $\mathbb{S}_H$, and let $U_B = U \setminus U_H$. Corollary 8.28 says that U_H is the overwhelmingly larger part of U and hence the size of U_H must be very big. But U_H hides the blob subconfigurations in $\mathbb{S}_H$ very much in a similar way as for hiding sets in the standard black-white pebble game. And we know from Section 7.3 that such sets need not be very large. Therefore we want to use Klawe-like ideas to derive a contradiction by transforming U_H locally into a (much) better blocking set for $\mathbb{S}_H$. The problem is that this might leave some subconfigurations in $\mathbb{S}_B$ not being blocked any longer (note that in general U_B will *not* on its own block $\mathbb{S}_B$). However, we have chosen our parameter $K = 13$ for the generalized LHC property (Property 8.7) very generously and the transformation in Section 7.3 works even for the (non-generalized) LHC property with parameter 1. Therefore, we expect our locally transformed blocking set to be so much cheaper that we can afford to take care of any subconfigurations in $\mathbb{S}_B$ that are no longer blocked simply by adding all bottom vertices for all black blobs in these subconfigurations to the blocking set.

We will not be able to pull this off by just making one local improvement of the hiding set as was done in Section 7.3, though. The reason is that the local improvement to U_H could potentially be very small, but lead to very many subconfigurations in $\mathbb{S}_B$ becoming unblocked. If so, we cannot afford adding new vertices blocking these subconfigurations without risking to increase the size and/or potential of our new blocking set too much. To make sure that this does not happen, we instead make multiple local improvements of U_H simultaneously. Our next lemma says that we can do this without losing control of how the measure behaves.

Lemma 8.30 (Generalization of Lemma 7.30)**.** *Suppose that $U_1, \ldots, U_k, V_1, \ldots, V_k, Y$ are vertex sets in a layered graph such that for all $i, j \in [k]$, $i \neq j$, it holds that $U_i \precsim_m V_i$, $V_i \cap V_j = \emptyset$, $U_i \cap V_j = \emptyset$ and $Y \cap V_i = \emptyset$. Then $m\left(Y \cup \bigcup_{i=1}^k U_i\right) \leq m\left(Y \cup \bigcup_{i=1}^k V_i\right)$.*

Proof. By induction over k. The base case $k = 1$ is Lemma 7.30 on page 522.

For the induction step, let $Y' = Y \cup \bigcup_{i=1}^{k-1} U_i$. Since $U_k \precsim_m V_k$ and $Y' \cap V_k = \emptyset$ by assumption, we get from Lemma 7.30 that

$$m\big(Y \cup \textstyle\bigcup_{i=1}^{k} U_i\big) = m\big(Y' \cup U_k\big) \leq m\big(Y' \cup V_k\big) = m\big(Y \cup \bigcup_{i=1}^{k-1} U_i \cup V_k\big) . \tag{8.22}$$

Letting $Y'' = Y \cup V_k$, we see that (again by assumption) it holds for all $i, j \in [k-1]$, $i \neq j$, that $U_i \precsim_m V_i$, $V_i \cap V_j = \emptyset$, $U_i \cap V_j = \emptyset$ and $Y'' \cap V_i = \emptyset$. Hence, by the induction hypothesis we have

$$m\big(Y \cup \textstyle\bigcup_{i=1}^{k-1} U_i \cup V_k\big) = m\big(Y'' \cup \bigcup_{k=1}^{i-1} U_i\big) \leq m\big(Y'' \cup \bigcup_{k=1}^{i-1} V_i\big) = m\big(Y \cup \bigcup_{i=1}^{k} V_i\big) \tag{8.23}$$

and the lemma follows. □

We also need an observation about the white pebbles in $\mathbb{S}_H$.

Observation 8.31. *For any $[B]\langle W \rangle \in \mathbb{S}_H$ with $b = \mathrm{bot}(B)$ it holds that $W = W \cap G^b_\triangle$.*

Proof. This is so since $\mathbb{S}$ is $\mathcal{W}$-eliminated with respect to U. Since $U \cup W$ hides $b = \mathrm{bot}(B)$, any vertices in $W \cap G^\triangledown_b$ are superfluous and will be removed by the $\mathcal{W}$-elimination procedure in Definition 8.15. □

Recalling from (8.13) that $\mathcal{W}^\triangle_H = \big\{W \cap G^b_\triangle \,\big|\, [B]\langle W \rangle \in \mathbb{S}_H, b = \mathrm{bot}(B)\big\}$ this leads to the next, simple but crucial observation.

Observation 8.32. *The vertex set $U_H \cup \mathcal{W}^\triangle_H$ hides the vertices in $\mathcal{B}_H$ in the sense of Definition 7.6.*

That is, we can consider $\big(\mathcal{B}_H, \mathcal{W}^\triangle_H\big)$ to be (almost)[7] a standard black-white pebble configuration. This sets the stage for applying the machinery of Section 7.3.

Appealing to Lemma 7.18 on page 517, let $X \subseteq U_H \,\dot\cup\, \mathcal{W}^\triangle_H$ be the unique, minimal tight set such that

$$\llbracket X \rrbracket = \llbracket U_H \,\dot\cup\, \mathcal{W}^\triangle_H \rrbracket \tag{8.24}$$

and define

$$\mathcal{W}^\triangle_T = \mathcal{W}^\triangle_H \cap X \tag{8.25a}$$

$$U_T = U_H \cap X \tag{8.25b}$$

to be the vertices in $\mathcal{W}^\triangle_H$ and U_H that remains in X after having applied the bottom-up pruning procedure of Lemma 7.18.

Let $\mathcal{H} = \mathcal{H}(G, X)$ be the hiding set graph of Definition 7.20 for $X = U_T \,\dot\cup\, \mathcal{W}^\triangle_T$. Suppose that $V_1, \ldots, V_k$ are the connected components of $\mathcal{H}$, and define for $i = 1, \ldots, k$ the vertex sets

$$\mathcal{B}^i_H = \mathcal{B}_H \cap V_i \tag{8.26a}$$

$$\mathcal{W}^i_H = \mathcal{W}^\triangle_H \cap V_i \tag{8.26b}$$

$$U^i_H = U_H \cap V_i \tag{8.26c}$$

[7]Not quite, since we might have $\mathcal{B}_H \cap \mathcal{W}^\triangle_H \neq \emptyset$. But at least we know that $U_H \cap \mathcal{W}^\triangle_H = \emptyset$ by $\mathcal{W}$-elimination and the roles of U and W in $U \cup W$ are fairly indistinguishable in Klawe's proof anyway, so this does not matter.

to be the black, white and "hiding" vertices within component V_i, and

$$\mathcal{W}_T^i = \mathcal{W}_T^{\triangle} \cap V_i \tag{8.26d}$$
$$U_T^i = U_T \cap V_i \tag{8.26e}$$

to be the vertices of $\mathcal{W}_H^{\triangle}$ and U_H in component V_i that "survived" when moving to the tight subset X. Note that we have the disjoint union equalities $\mathcal{W}_H^{\triangle} = \dot{\bigcup}_{i=1}^{k} \mathcal{W}_H^i$, $U_H = \dot{\bigcup}_{i=1}^{k} U_H^i$, et cetera for all of these sets.

Let us also generalize Definition 7.8 of measure and partial measure to multi-sets of vertices in the natural way, where we charge separately for each copy of every vertex. This is our way of doing the bookkeeping for the extra vertices that might be needed later to block $\mathbb{S}_B$ in the final step of our construction.

This brings us to the key lemma stating how we will locally improve the blocking sets.

Lemma 8.33 (Generalization of Lemma 7.36)**.** *With the assumptions on the blob-pebbling configuration $\mathbb{S}$ and the vertex set U as in Lemma 8.26 and with notation as above, suppose that $U_H^i \cup \mathcal{W}_H^i$ hides $\mathcal{B}_H^i$, that $\mathcal{H}\left(U_T^i \cup \mathcal{W}_T^i\right)$ is a connected graph, and that*

$$\left|U_H^i\right| \geq 6 \cdot \left|\mathcal{B}_H^i \cup \mathcal{W}_H^i\right| . \tag{8.27}$$

Then we can find a multi-set $U_*^i \subseteq \llbracket U_T^i \cup \mathcal{W}_T^i \rrbracket$ that hides the vertices in $\mathcal{B}_H^i$, has $\left\lfloor |U_H^i|/3 \right\rfloor$ extra copies of some fixed but arbitrary vertex on level $L_U = \operatorname{maxlevel}\left(U_H^i\right)$, and satisfies $U_*^i \precsim_m U_H^i$ and $\left|U_*^i\right| < \left|U_H^i\right|$ (where U_*^i is measured and counted as a multi-set with repetitions).

Proof. Let U_*^i be the set found in Lemma 7.33 on page 523, which certainly is in $\llbracket U_T^i \cup \mathcal{W}_T^i \rrbracket$, together with the prescribed extra copies of some (fixed but arbitrary) vertex that we place on level $\operatorname{maxlevel}\left(\llbracket U_H^i \cup \mathcal{W}_H^i \rrbracket\right) \geq L_U$ to be on the safe side. By Lemma 7.33, U_*^i hides $\mathcal{B}_H^i$, and the size of U_*^i *counted as a multi-set with repetitions* is

$$\left|U_*^i\right| \leq \left|\mathcal{B}_H^i\right| + \left\lfloor |U_H^i|/3 \right\rfloor \leq \left(\tfrac{1}{6} + \tfrac{1}{3}\right) \cdot \left|U_H^i\right| < \left|U_H^i\right| . \tag{8.28}$$

It remains to show that $U_*^i \precsim_m U_H^i$.

The proof of this last measure inequality is very much as in Lemma 7.36, but with the difference that the connected graph that we are dealing with is defined over $U_T^i \,\dot{\cup}\, \mathcal{W}_T^i$, but we count the vertices in $U_H^i \,\dot{\cup}\, \mathcal{W}_H^i$. Note, however, that by construction these two unions hide exactly the same set of vertices, i. e.,

$$\llbracket U_T^i \,\dot{\cup}\, \mathcal{W}_T^i \rrbracket = \llbracket U_H^i \,\dot{\cup}\, \mathcal{W}_H^i \rrbracket . \tag{8.29}$$

Recall that by Definition 7.29 on page 522, what we need to do in order to show that $U_*^i \precsim_m U_H^i$ is to find for each j an $l \leq j$ such that $m^j\left(U_*^i\right) \leq m^l\left(U_H^i\right)$. As in Lemma 7.36, we divide the proof into two cases.

1. If $j \leq \text{minlevel}\big(U_T^i \cup \mathcal{W}_T^i\big) = \text{minlevel}\big(U_H^i \cup \mathcal{W}_H^i\big)$, we get

$$\begin{aligned}
m^j\big(U_*^i\big) &= j + 2 \cdot \big|U_*^i\{\succeq j\}\big| && \big[\text{ by definition of } m^j(\cdot)\ \big] \\
&\leq j + 2 \cdot \big|U_*^i\big| && \big[\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V\ \big] \\
&\leq j + 2 \cdot \big(|\mathcal{B}_H^i| + \big\lfloor |U_H^i|/3 \big\rfloor\big) && \big[\text{ by Lemma 7.33 plus extra vertices }\big] \\
&< j + 2 \cdot \big|U_H^i\big| && \big[\text{ by the assumption in (8.27) }\big] \\
&= j + 2 \cdot \big|U_H^i\{\succeq j\}\big| && \big[\ U_H^i\{\succeq j\} = U_H^i \text{ since } j \leq \text{minlevel}(U_H^i)\ \big] \\
&= m^j(U_H^i) && \big[\text{ by definition of } m^j(\cdot)\ \big]
\end{aligned}$$

and we can choose $l = j$ in Definition 7.29.

2. Consider instead $j > \text{minlevel}\big(U_T^i \cup \mathcal{W}_T^i\big)$ and let $L = \text{minlevel}\big(U_T^i \cup \mathcal{W}_T^i\big)$. Since the black pebbles in $\mathcal{B}_H^i$ are hidden by $U_T^i \cup \mathcal{W}_T^i$, i. e., $\mathcal{B}_H^i \subseteq \llbracket U_T^i \cup \mathcal{W}_T^i \rrbracket$ in formal notation, recollecting Definition 7.31 and Observation 7.32, part 2, we see that

$$L_{\succeq j}\big(\mathcal{B}_H^i\big) \leq L_{\succeq j}\big(\llbracket U_T^i \cup \mathcal{W}_T^i \rrbracket\big) \tag{8.30}$$

for all j. Also, since $U_T^i \cup \mathcal{W}_T^i$ is a hiding-connected vertex set in a spreading graph G, combining Definition 7.34 with the fact that $U_T^i \cup \mathcal{W}_T^i \subseteq U_H^i \cup \mathcal{W}_H^i$ we can derive that

$$j + L_{\succeq j}\big(\llbracket U_T^i \cup \mathcal{W}_T^i \rrbracket\big) \leq L + \big|U_T^i \cup \mathcal{W}_T^i\big| \leq L + \big|U_H^i \cup \mathcal{W}_H^i\big|\,. \tag{8.31}$$

Together, (8.30) and (8.31) say that

$$j + L_{\succeq j}\big(\mathcal{B}_H^i\big) \leq L + \big|U_H^i \cup \mathcal{W}_H^i\big| \tag{8.32}$$

and using this inequality we can show that

$$\begin{aligned}
m^j(U_*^i) &= j + 2 \cdot \big|U_*^i\{\succeq j\}\big| && \big[\text{ by definition of } m^j(\cdot)\ \big] \\
&\leq j + L_{\succeq j}\big(\mathcal{B}_H^i\big) + \big|\mathcal{B}_H^i\big| + 2 \cdot \big\lfloor |U_H^i|/3 \big\rfloor && \big[\text{ by Lemma 7.33 + extra vertices }\big] \\
&\leq L + \big|U_H^i \cup \mathcal{W}_H^i\big| + \big|\mathcal{B}_H^i\big| + 2 \cdot \big\lfloor |U_H^i|/3 \big\rfloor && \big[\text{ using the inequality (8.32) }\big] \\
&\leq L + \tfrac{5}{3}\big|U_H^i\big| + \big|\mathcal{B}_H^i\big| + \big|\mathcal{W}_H^i\big| && \big[\ |A \cup B| \leq |A| + |B|\ \big] \\
&\leq L + \tfrac{5}{3}\big|U_H^i\big| + 2 \cdot \big|\mathcal{B}_H^i \cup \mathcal{W}_H^i\big| && \big[\ |A| + |B| \leq 2 \cdot |A \cup B|\ \big] \\
&\leq L + 2 \cdot \big|U_H^i\big| && \big[\text{ by the assumption in (8.27) }\big] \\
&= L + 2 \cdot |U_H^i\{\succeq L\}| && \big[\text{ since } L \leq \text{minlevel}(U_H^i)\ \big] \\
&= m^L(U_H^i) && \big[\text{ by definition of } m^L(\cdot)\ \big]\,.
\end{aligned}$$

Thus, the partial measure of U_H^i at the minimum level L is always at least as large as the partial measure of U_*^i at levels j above this minimum level, and we can choose $l = L$ in Definition 7.29.

Consequently, $U_*^i \precsim_m U_H^i$ and the lemma follows. □

Now we want to determine in which connected components of the hiding set graph $\mathcal{H}$ we should apply Lemma 8.33. Loosely put, we want to be sure that changing U_H^i to U_*^i is worthwhile, i. e., that we gain enough from this transformation to compensate for the extra hassle of reblocking blobs in $\mathbb{S}_B$ that turn unblocked when we change U_H^i. With this in mind, let us define the *weight* of a component V_i in $\mathcal{H}$ as

$$w(V_i) = \begin{cases} \lceil |U_H^i|/6 \rceil & \text{if } \left|U_H^i\right| \geq 6 \cdot \left|\mathcal{B}_H^i \cup \mathcal{W}_H^i\right|, \\ 0 & \text{otherwise.} \end{cases} \tag{8.33}$$

The idea is that a component V_i has large weight if the hiding set U_H^i in this component is large compared to the number of bottom black vertices in $\mathcal{B}_H^i$ hidden and the white pebbles $\mathcal{W}_H^i$ helping U_H^i to hide $\mathcal{B}_H^i$. If we concentrate on changing the hiding sets in components with non-zero weight, we hope to gain more from the transformation of U_H^i into U_*^i than we lose from then having to reblocking $\mathbb{S}_B$. And since U_H is large, the total weight of the non-zero-weight components is guaranteed to be reasonably large.

Proposition 8.34. *With notation as above, the total weight of all connected components $V_1, \ldots, V_k$ in the hiding set graph $\mathcal{H} = \mathcal{H}\big(G, U_T \cup \mathcal{W}_T^{\triangle}\big)$ is $\sum_{i=1}^{k} w(V_i) > \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|$.*

Proof. The total size of the union of all subsets $U_H^i \subseteq U_H$ with sizes $\left|U_H^i\right| < 6 \cdot \left|\mathcal{B}_H^i \cup \mathcal{W}_H^i\right|$ resulting in zero-weight components V_i in $\mathcal{H}$ is clearly strictly less than

$$6 \cdot \sum_{i=1}^{k} \left|\mathcal{B}_H^i \cup \mathcal{W}_H^i\right| = 6 \cdot \left|\mathcal{B}_H \cup \mathcal{W}_H^{\triangle}\right| \leq 6 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|. \tag{8.34}$$

Since according to Corollary 8.28 we have that $\left|U_H\right| \geq 12 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|$, it follows that the size of the union $\bigcup_{w(V_i)>0} U_H^i$ of all subsets U_H^i corresponding to non-zero-weight components V_i must be strictly larger than $6 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right|$. But then

$$\sum_{w(V_i)>0} w(V_i) \geq \sum_{w(V_i)>0} \lceil |U_H^i|/6 \rceil \geq \frac{1}{6} \cdot \left| \bigcup_{w(V_i)>0} U_H^i \right| > \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right| \tag{8.35}$$

as claimed in the proposition. □

We have now collected all tools needed to establish the generalized limited hiding-cardinality property for spreading graphs. Before we wrap up the proof, let us recapitulate what we have shown so far.

We have divided the blocking set U into a disjoint union $U_H \dot\cup U_B$ of the vertices U_H not only blocking but actually *hiding* the subconfigurations in $\mathbb{S}_H \subseteq \mathbb{S}$, and the vertices U_B just helping U_H to block the remaining subconfigurations in $\mathbb{S}_B = \mathbb{S} \setminus \mathbb{S}_H$. In Lemma 8.27 and Corollary 8.28, we proved that if U is large (which we are assuming) then U_B must be very small compared to U_H, so we can basically just ignore U_B. If we want to do something interesting, it will have to be done with U_H.

And indeed, Lemma 8.33 tells us that we can restructure U_H to get a new vertex set hiding $\mathbb{S}_H$ and make considerable savings, but that this can lead to $\mathbb{S}_B$ no longer being blocked. By Proposition 8.34, there is a large fraction of U_H that resides in the non-zero-weight components of the hiding set graph $\mathcal{H}$ (as defined in equation (8.33)). We would like to show that by judiciously performing the restructuring of Lemma 8.33 in these components, we can also take care of $\mathbb{S}_B$.

More precisely, we claim that we can combine the hiding sets U^i_* from Lemma 8.33 with some subsets of $U_H \cup U_B$ and $\mathcal{B}_B$ into a new blocking set U^* for all of $\mathbb{S}_H \cup \mathbb{S}_B = \mathbb{S}$ in such a way that the measure $m(U^*)$ does not exceed $m(U) = \mathrm{pot}(\mathbb{S})$ but so that $|U^*| < |U|$. But this contradicts the assumptions in Lemma 8.26. It follows that the conclusion in Lemma 8.26, which we assumed to be false in order to derive a contradiction, must instead be true. That is, any set U that is chosen as in Lemma 8.26 must have size $|U| \leq 13 \cdot |\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}^{\triangle}_H|$. This in turn implies Theorem 8.19, i. e., that layered spreading graphs possess the generalized limited hiding-cardinality property that we assumed in order to get a lower bound on blob-pebbling price, and we are done.

We proceed to establish this final claim. Our plan is once again to do some bipartite matching with the help of Hall's marriage theorem. To this end, create a weighted bipartite graph with the vertices in $\mathcal{B}_B = \{\mathrm{bot}(B) \,|\, [B]\langle W\rangle \in \mathbb{S}_B\}$ on the left-hand side and with the non-zero-weight connected components among $V_1, \ldots, V_k$ in $\mathcal{H}$ in the sense of (8.33) acting as "supervertices" on the right-hand side. Reorder the indices among the connected components $V_1, \ldots, V_k$ if needed so that the non-zero-weight components are $V_1, \ldots, V_{k'}$. All vertices in the weighted graphs are assigned weights so that each right-hand side supervertex V_i gets its weight according to (8.33), and each left-hand vertex has weight 1.[8] We define the neighbours of each fixed vertex $b \in \mathcal{B}_B$ to be

$$N(b) = \{V_i \,|\, w(V_i) > 0 \text{ and } \mathrm{maxlevel}(U^i_H) > \mathrm{level}(b)\}\,, \tag{8.36}$$

i. e., all non-zero-weight components V_i that contain vertices in the hiding set U_H that could possibly be involved in blocking any subconfiguration $[B]\langle W\rangle \in \mathbb{S}_B$ having bottom vertex $\mathrm{bot}(B) = b$. This is so since by Proposition 8.24, any vertex $u \in U_H$ helping to block such a subconfiguration $[B]\langle W\rangle \in \mathbb{S}_B$ must be strictly above b, so if the highest-level vertices in U^i_H are on a level below b, no vertex in U^i_H can be responsible for blocking $[B]\langle W\rangle$.

Let $\mathcal{B}' \subseteq \mathcal{B}_B$ be a largest set such that $w(N(\mathcal{B}')) \leq |\mathcal{B}'|$. We must have

$$N(\mathcal{B}') \neq \textstyle\bigcup_{i=1}^{k'} V_i \tag{8.37}$$

since $w(\bigcup_{i=1}^{k'} V_i) > |\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}^{\triangle}_H| \geq |\mathcal{B}_B|$ by Proposition 8.34. For all $\mathcal{B}'' \subseteq \mathcal{B}_B \setminus \mathcal{B}'$ it holds that

$$w(N(\mathcal{B}'') \setminus N(\mathcal{B}')) \geq |\mathcal{B}''| \tag{8.38}$$

since otherwise $\mathcal{B}'$ would not be of largest size as assumed above. The inequality (8.38) plugged into Hall's marriage theorem tells us that there is a matching of the vertices in $\mathcal{B}_B \setminus \mathcal{B}'$ to the components in $\bigcup_{i=1}^{k'} V_i \setminus N(\mathcal{B}') \neq \emptyset$ with the property that no component V_i gets matched with more than $w(V_i)$ vertices from $\mathcal{B}_B \setminus \mathcal{B}'$.

Reorder the components in the hiding set graph $\mathcal{H}$ so that the matched components are $V_1, \ldots, V_m$ and the rest of the components are $V_{m+1}, \ldots, V_k$ and so that $U^1_H, \ldots, U^m_H$ and $U^{m+1}_H, \ldots, U^k_H$ are the corresponding subsets of the hiding set U_H. Then pick good local blockers $U^i_* \subseteq V_i$ as in Lemma 8.33 for all components $V_1, \ldots, V_m$. Now the following holds:

1. By construction and assumption, respectively, the vertex set $\bigcup_{i=1}^{m} U^i_* \cup \bigcup_{i=m+1}^{k} U^i_H$ blocks (and even hides) $\mathbb{S}_H$.

[8] Or, if we like, we can equivalently think of an unweighted graph, where each V_i is a cloud of $w(V_i)$ unique and distinct vertices, and where $N(b)$ in (8.36) always containing either all or none of these vertices.

2. All subconfigurations in
$$\mathbb{S}_B^1 = \left\{[B]\langle W\rangle \in \mathbb{S}_B \,\middle|\, \mathrm{bot}(B) \in \mathcal{B}'\right\} \tag{8.39}$$
are blocked by $U_B \cup N\bigl(\mathcal{B}'\bigr) = U_B \cup \bigcup_{i=m+1}^{k} U_H^i$, as we have not moved any elements in U above $\mathcal{B}'$.

3. With notation as in Lemma 8.30, let $Y = U_B \cup \bigcup_{i=m+1}^{k} U_H^i$ and consider U_*^i and U_H^i for $i = 1, \ldots, m$. We have $U_*^i \precsim_m U_H^i$ for $i = 1, \ldots, m$ by Lemma 8.33. Also, since $U_H \cap U_B = \emptyset$ and $U_*^i \subseteq V_i$ and $U_H^i \subseteq V_i$ for $V_1, \ldots, V_k$ pairwise disjoint sets of vertices, it holds for all $i, j \in [m]$, $i \neq j$, that $U_*^i \cap U_*^j = \emptyset$, $U_H^i \cap U_H^j = \emptyset$, $U_*^i \cap U_H^j = \emptyset$ and $Y \cap U_H^j = \emptyset$. Therefore, the conditions in Lemma 8.30 are satisfied and we conclude that
$$\begin{aligned} m\bigl(U_B \cup \textstyle\bigcup_{i=1}^{m} U_*^i \cup \bigcup_{i=m+1}^{k} U_H^i\bigr) &= m\bigl(Y \cup \textstyle\bigcup_{i=1}^{m} U_*^i\bigr) \\ &\leq m\bigl(Y \cup \textstyle\bigcup_{i=1}^{m} U_H^i\bigr) \\ &= m\bigl(U_B \cup \textstyle\bigcup_{i=1}^{m} U_H^i \cup \bigcup_{i=m+1}^{k} U_H^i\bigr) \\ &= m(U)\,, \end{aligned} \tag{8.40}$$
where we note that $U_B \cup \bigcup_{i=1}^{m} U_*^i \cup \bigcup_{i=m+1}^{k} U_H^i$ is *measured as a multi-set with repetitions*. Also, we have the strict inequality
$$\left|U_B \cup \textstyle\bigcup_{i=1}^{m} U_*^i \cup \bigcup_{i=m+1}^{k} U_H^i\right| < |U|\,, \tag{8.41}$$
where again the multi-set is *counted with repetitions*.

4. It remains to take care of the potentially unblocked subconfigurations in
$$\mathbb{S}_B^2 = \left\{[B]\langle W\rangle \in \mathbb{S}_B \,\middle|\, \mathrm{bot}(B) \in \mathcal{B}_B \setminus \mathcal{B}'\right\}. \tag{8.42}$$
But we derived above that there is a matching of $\mathcal{B}_B \setminus \mathcal{B}'$ to $V_1, \ldots, V_m$ such that no V_i is chosen by more than
$$w(V_i) = \left\lceil |U_H^i|/6 \right\rceil \leq \left\lfloor |U_H^i|/3 \right\rfloor \tag{8.43}$$
vertices from $\mathcal{B}_B \setminus \mathcal{B}'$ (where we used that $\left|U_H^i\right| \geq 6$ if $w(V_i) > 0$ to get the last inequality). This means that there is a spare blocker vertex in U_*^i for each $b \in \mathcal{B}_B \setminus \mathcal{B}'$ that is matched to V_i. Also, by the definition of neighbours in our weighted bipartite graph, each b is matched to a component with $\mathrm{maxlevel}\bigl(U_H^i\bigr) > \mathrm{level}(b)$. By Observation 8.29, lowering these spare vertices from $\mathrm{maxlevel}\bigl(U_H^i\bigr)$ to $\mathrm{level}(b)$ can only decrease the measure.

Finally, throw away any remaining multiple copies in our new blocking set, and denote the resulting set by U^*. We have that U^* blocks $\mathbb{S}$ and that $m\bigl(U^*\bigr) \leq m(U)$ but $\left|U^*\right| < |U|$. This is a contradiction since U was chosen to be of minimal size, and thus Lemma 8.26 must hold. But then Theorem 8.19 follows immediately as well, as was noted above.

8.5 Recapitulation of the proof of Theorem 1.1 and optimality of result

Let us conclude this section by recalling why the tight bound on clause space for refuting pebbling contradictions in Theorem 1.1 now follows and by showing that the current construction cannot be pushed to give a better result.

To prove the upper bound in the main theorem, we need the fact that the black pebbling price of G provides an upper bound for the refutation clause space of Peb^d_G. Let us state this formally as a proposition and provide a proof for completeness.

Proposition 8.35. *For any DAG G with vertex indegrees 0 or 2, $Sp(Peb^d_G \vdash \perp) \leq \mathsf{Peb}(G)+4$.*

Proof. Given a black pebbling of G, we construct a refutation of Peb^d_G such that if at some point in time there are black pebbles on a set of vertices V, then we have the clauses $\left\{\bigvee_{i=1}^d x(v)_i \mid v \in V\right\}$ in memory. When some new vertex v is pebbled, we derive $\bigvee_{i=1}^d x(v)_i$ from the clauses already in memory. We claim that with a little care, this can be done in constant extra space independent of d. When a black pebble is removed from v, we erase the clause $\bigvee_{i=1}^d x(v)_i$. We conclude the resolution proof by resolving $\bigvee_{i=1}^d x(z)_i$ for the target z with all target axioms $\overline{x(z)}_i$, $i \in [d]$, in space 3.

It is clear that given our claim about the constant extra space needed when a vertex is black-pebbled, this yields a resolution refutation in space equal to the pebbling cost plus some constant. In particular, given an optimal black pebbling of G, we get a refutation in space $\mathsf{Peb}(G)+\mathrm{O}(1)$.

To prove the claim, note first that it trivially holds for source vertices v, since $\bigvee_{i=1}^d x(v)_i$ is an axiom of the formula. Suppose for a non-source vertex r with predecessors p and q that at some point in time a black pebble is placed on r. Then p and q must be black-pebbled, so by induction we have the clauses $\bigvee_{i=1}^d x(p)_i$ and $\bigvee_{j=1}^d x(q)_j$ in memory. We will use that the clause $\overline{x(p)}_i \vee \bigvee_{l=1}^d x(r)_l$ for any i can be derived in additional space 3 by resolving $\bigvee_{j=1}^d x(q)_j$ with $\overline{x(p)}_i \vee \overline{x(q)}_j \vee \bigvee_{l=1}^d x(r)_l$ for $j \in [d]$, leaving the easy verification of this fact to the reader. To derive $\bigvee_{l=1}^d x(r)_l$, first resolve $\bigvee_{i=1}^d x(p)_i$ with $\overline{x(p)}_1 \vee \bigvee_{l=1}^d x(r)_l$ to get $\bigvee_{i=2}^d x(p)_i \vee \bigvee_{l=1}^d x(r)_l$, and then resolve this clause with the clauses $\overline{x(p)}_i \vee \bigvee_{l=1}^d x(r)_l$ for $i = 2,\ldots,d$ one by one to get $\bigvee_{l=1}^d x(r)_l$ in total extra space 4. ☐

Theorem 8.36 (rephrasing of Theorem 1.1)**.** *Suppose that G_h is a layered DAG of height h that is spreading. Then the clause space of refuting the pebbling contradiction $Peb^d_{G_h}$ of degree $d > 1$ by resolution is $Sp(Peb^d_{G_h} \vdash \perp) = \Theta(h)$.*

Proof. The $\mathrm{O}(h)$ upper bound on clause space follows from the bound $\mathsf{Peb}(G_h) \leq h+\mathrm{O}(1)$ on the black pebbling price in Lemma 7.2 on page 509 combined with the bound $Sp(Peb^d_G \vdash \perp) \leq \mathsf{Peb}(G)+\mathrm{O}(1)$ from Proposition 8.35.

For the lower bound, we instead consider the pebbling formula $*Peb^d_{G_h}$ with the target axioms $\overline{x(z)}_1,\ldots,\overline{x(z)}_d$ removed, and use that by Lemma 5.1 on page 495 it holds that $Sp\left(Peb^d_{G_h} \vdash \perp\right) = Sp\left(*Peb^d_{G_h} \vdash \bigvee_{i=1}^d x(z)_i\right)$. Fix any resolution derivation $\pi : *Peb^d_{G_h} \vdash \bigvee_{i=1}^d x(z)_i$ and let $\mathcal{P}_\pi$ be the complete blob-pebbling of the graph G associated to π in Theorem 5.3 on page 496 such that $\mathsf{cost}(\mathcal{P}_\pi) \leq \max_{\mathbb{C}\in\pi}\left\{\mathsf{cost}(\mathbb{S}(\mathbb{C}))\right\}+\mathrm{O}(1)$. On the one hand, Theorem 6.5 on page 505 says that $\mathsf{cost}(\mathbb{S}(\mathbb{C})) \leq |\mathbb{C}|$ provided that $d > 1$, so in particular it must hold that $\mathsf{cost}(\mathcal{P}_\pi) \leq Sp(\pi)+\mathrm{O}(1)$. On the other hand, $\mathsf{cost}(\mathcal{P}_\pi) \geq \mathsf{Blob\text{-}Peb}(G_h)$ by definition, and by Theorems 8.8 and 8.19 it holds that $\mathsf{Blob\text{-}Peb}(G_h) = \Omega(h)$. Thus $Sp(\pi) = \Omega(h)$, and the theorem follows. ☐

Plugging in pyramid graphs Π_h in Theorem 8.36, we get 4-CNF formulas F_n of size $\Theta(n)$ with refutation clause space $\Theta(\sqrt{n})$. This is the best we can get from pebbling formulas over spreading graphs.

Theorem 8.37. *Let G be any layered spreading graph and suppose that Peb^d_G has formula size and number of clauses $\Theta(n)$. Then $Sp\left(Peb^d_G \vdash \perp\right) = \mathrm{O}(\sqrt{n})$.*

Proof. Suppose that G has height h. Then $Sp\left(Peb^d_G \vdash \perp\right) = \mathrm{O}(h)$ as was noted above. The size of Peb^d_G, as well as the number of clauses, is linear in the number of vertices $|V(G)|$. We claim that the fact that G is spreading implies that $|V(G)| = \Omega\left(h^2\right)$, from which the theorem follows.

To prove the claim, let V_L denote the vertices of G on level L. Then $|V(G)| = \sum_{L=0}^{h} |V_L|$. Obviously, for any L the set V_L hides the sink z of G. Fix for every L some arbitrary minimal subset $V'_L \subseteq V_L$ hiding z. Then V'_L is tight, the graph $\mathcal{H}(V'_L)$ is hiding-connected by Corollary 7.23, and setting $j = h$ in the spreading inequality (7.10) we get that $\left|V'_L\right| \geq 1 + h - L$. Hence $|V(G)| \geq \sum_{L=0}^{h} |V'_L| = \Omega\left(h^2\right)$. $\square$

The proof of Theorem 8.37 can also be extended to cover the original definition in [42] of spreading graphs that are not necessarily layered, but we omit the details.

9 Conclusion and open problems

We have proven an asymptotically tight bound on the refutation clause space in resolution of pebbling contradictions over pyramid graphs, improving on previously known separations of length and clause space in resolution. Also, in contrast to previous polynomial lower bounds on clause space, our result does not not follow from lower bounds on width for the same formulas. Instead, a corollary of our result is an exponential improvement of the separation of width and space in [47]. In retrospect, the current paper can be seen as a crucial step in settling the question of the relationship between length and space discussed in, e. g., [12, 32, 59, 60], and subsequently resolved in [15].

More technically speaking, we have established that for all graphs G in the class of "layered spreading DAGs" (including complete binary trees and pyramid graphs) the height h of G, which coincides with the black-white pebbling price, is an asymptotic lower bound for the refutation clause space $Sp\left(Peb^d_G \vdash \perp\right)$ of pebbling contradictions Peb^d_G provided that $d \geq 2$. Plugging in pyramid graphs we get a $\Theta\left(\sqrt{n}\right)$ bound on space, which is the best one can get for any spreading graph.

An obvious question is whether this lower bound on clause space in terms of black-white pebbling price is true for arbitrary DAGs. In particular, does it hold for the family of DAGs $\{G_n\}_{n=1}^{\infty}$ in [35] of size $\mathrm{O}(n)$ that have maximal black-white pebbling price $\textit{BW-Peb}(G_n) = \Theta(n/\log n)$ in terms of size? We conjecture that this is indeed the case.

Conjecture 9.1. *Let G be any DAG with constant fan-in and a single sink, and let Peb^d_G denote the pebbling contradiction of degree $d > 1$ defined over G. Then the clause space of refuting Peb^d_G in resolution is $Sp(Peb^d_G \vdash \perp) \geq \textit{BW-Peb}(G)$.*

If this conjecture can be shown to be true, then this would also reprove the asymptotically optimal length-space separation in [15] with better constant factors.

A natural approach towards proving Conjecture 9.1 would be to extend the lower bounds on pebbling price in the blob-pebble game used in this paper to the graphs in [35]. It seems challenging to lift the lower bound argument in [35] from standard black-white pebbling to blob-pebbling, however. While it appears plausible that the same lower bound should hold, we have not been able to prove this.

If this line of attack could be made to work, it could also be a promising approach towards establishing the PSPACE-completeness of space requirements for k-CNF formulas in resolution, perhaps building on the constructions in [34, 38] (as noted before, this problem is easily seen to be in PSPACE).

Open Problem 9.2. *Given a k-CNF formula F and a space bound s, is it* PSPACE*-complete to determine whether F can be refuted in resolution in clause space s? And is it* PSPACE*-complete to decide whether F can be refuted in total space s?*

The idea here would be to reduce from the problem of deciding the pebbling price of a graph G to the problem of deciding the space requirements for the pebbling formula over G. We remark that it seems unlikely that the techniques in [15] could be used to resolve this question. The reason for this is that there is an inherent blow-up of a factor at least 2 between the upper and lower bounds one gets for the version of pebbling formulas used in that paper, and therefore one would need a constant-factor PSPACE non-approximability gap in the pebbling problem in order to get a meaningful reduction to refutation space. In contrast, for our pebbling formulas it seems that the true upper and lower bounds should be within a small additive constant of one another, or should perhaps even coincide. As noted above, however, the lower bound proven in the current paper appears to be off by a constant multiplicative factor, and additional ideas seem to be needed to obtain an exact bound.

Returning to the discussion in the introductory section, a third open question is as follows. We know from [17] that short resolution refutations imply the existence of narrow refutations, and in view of this an appealing proof search heuristic is to search exhaustively for refutations in minimal width. One serious drawback of this approach is that there is no guarantee that the short and narrow refutations are the same one. On the contrary, the narrow refutation π' resulting from the proof in [17] is potentially exponentially longer than the short proof π that we start with. However, we have no examples of formulas where the refutation in minimum width is actually known to be substantially longer than the minimum-length refutation. Therefore, it would be very interesting to know whether this increase in length is necessary. That is, is there a formula family which exhibits a length-width trade-off in the sense that there are short refutations and narrow refutations, but all narrow refutations have a length blow-up (polynomial or superpolynomial)? Or is the exponential blow-up in [17] just an artifact of the proof?

Open Problem 9.3. *If F is a k-CNF formula over n variables refutable in length L, can one always find a refutation π of F in width $W(\pi) = \mathrm{O}\big(\sqrt{n \log L}\big)$ with length no more than, say, $L(\pi) = \mathrm{O}(L)$ or at most $\mathrm{poly}(L)$? Or is there a formula family which necessarily exhibits a length-width trade-off in the sense that there are short refutations and narrow refutations, but all narrow refutations have a length blow-up (polynomial or superpolynomial)?*

Similar questions can be posed for clause space and total space. Suppose that we have a formula that is refutable in both small space and short length. Is it then possible to refute the formula in small space and short length *simultaneously*, possibly increasing the space by a constant factor or the length by some polynomial factor? In recent joint work with Ben-Sasson [16], the first author gave a strong negative answer to this question by proving superpolynomial or even exponential trade-offs for length with respect to both total space and clause space for space in almost the whole range between constant and linear in the formula size. Even more recently, Beame et al. [9] established that although any formula can be refuted in linear clause space (as shown in [31]), there are formulas with refutations in superlinear but polynomial length and space where bringing the space down to linear incurs a superpolynomial length blow-up.

However, it still remains open what kind of trade-offs are possible for constant space. Given a formula refutable in constant clause space, we know that there must exist a refutation in polynomial length as well. This follows by applying the upper bound on width in terms of clause space from [5], and then noting that narrow proofs are trivially short. But as in [17], the refutation we end up with is not the same as the one with which we started. This leads to the following question.

Open Problem 9.4. *Given a family of polynomial-size k-CNF formulas* $\{F_n\}_{n=1}^{\infty}$ *with refutation clause space* $Sp(F_n \vdash \perp) = \mathrm{O}(1)$*, are there refutations* $\pi : F_n \vdash \perp$ *simultaneously in length* $L(\pi) = \mathrm{poly}(n)$ *and clause space* $Sp(\pi) = \mathrm{O}(1)$*? Or is it possible that restricting the space to constant can force the length to be superpolynomial?*

We conclude with a brief discussion on resolution space versus clause learning. Given that research into proof space complexity was originally motivated by SAT solving concerns, we find it an interesting question whether lower bounds on space can give indications as to which formulas might be hard for clause learning algorithms and why. On the face of it, space lower bounds might not seem too relevant since they will never be stronger than linear, and certainly any SAT solver will use linear space to store the formula in question. Suppose, however, that we know for some CNF formula that its refutation space is large. What this tells us is that any algorithm, even a non-deterministic one making optimal choices concerning which clauses to save or throw away at any given point in time, will have to keep a fairly large number of "active" clauses in memory in order to carry out the refutation. Since this is so, a real-life deterministic proof search algorithm, which has no sure-fire way of knowing which clauses are the right ones to concentrate on at any given moment, might have to keep working on a lot of extra clauses in order to be certain that the fairly large critical set of clauses needed to find a refutation will be among the "active" clauses. In this sense a (non-deterministic) linear lower bound might imply much stronger bounds in practice. We remark that indeed, resolution refutation space has been proposed as a measure of hardness for SAT solvers in [4].

And intriguingly enough, pebbling contradictions might in fact be an example of the phenomenon discussed in the previous paragraph. We know that these formulas are very easy with respect to length and width, having constant-width refutations that are essentially as short as the formulas themselves. But in [57], it was shown that state-of-the-art clause learning algorithms can have serious problems with even moderately large pebbling contradictions over pyramid graphs.[9] And more recent experimental work involving the first author and reported in [40] indicates that the space complexity of pebbling contradictions is correlated with how hard or easy the formulas are for CDCL solvers. Although we are certainly not arguing that this is the whole story—it was also shown in [57] that the branching order is a critical factor, and that given some extra structural information the algorithm can achieve an exponential speed-up, and there are some important caveats in the experimental results in [40]—we believe that such possible connections would be worth investigating further. It would also be interesting to investigate for

[9]The "grid pebbling formulas" in [57] are exactly our pebbling contradictions of degree $d = 2$ over pyramid graphs.

the formulas in [9, 16] whether anything similar to the theoretical size-space trade-offs for these formulas actually shows up as trade-offs between running time and memory consumption in practice.

Acknowledgements

We are thankful to Per Austrin and Mikael Goldmann for generous feedback during various stages of this work, and to Gunnar Kreitz for quickly spotting some bugs in a preliminary version of the blob-pebble game. Also, we would like to thank Paul Beame and Maria Klawe for valuable correspondence concerning their work, Nathan Segerlind for comments and pointers regarding clause learning, and Eli Ben-Sasson for stimulating discussions about proof complexity in general and the problems in Section 9 in particular. Finally, we gratefully acknowledge the thoughtful comments of the anonymous reviewers and the patient and meticulous work of the Theory of Computing editors, which helped us to improve the presentation considerably.

References

[1] RON AHARONI AND NATHAN LINIAL: Minimal non-two-colorable hypergraphs and minimal unsatisfiable formulas. *J. Combin. Theory Ser. A*, 43(2):196–204, 1986. [doi:10.1016/0097-3165(86)90060-9] 503

[2] MICHAEL ALEKHNOVICH, ELI BEN-SASSON, ALEXANDER A. RAZBOROV, AND AVI WIGDERSON: Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211, 2002. Preliminary version in STOC'00. [doi:10.1137/S0097539700366735] 476, 485, 486

[3] MICHAEL ALEKHNOVICH, JAN JOHANNSEN, TONIANN PITASSI, AND ALASDAIR URQUHART: An exponential separation between regular and general resolution. *Theory of Computing*, 3(5):81–102, 2007. Preliminary version in STOC'02. [doi:10.4086/toc.2007.v003a005] 486

[4] CARLOS ANSÓTEGUI, MARÍA LUISA BONET, JORDI LEVY, AND FELIP MANYÀ: Measuring the hardness of SAT instances. In *Proc. 23rd AAAI Conference on Artificial Intelligence (AAAI'08)*, pp. 222–228. AAAI Press, 2008. AAAI. [ACM:1620032] 550

[5] ALBERT ATSERIAS AND VÍCTOR DALMAU: A combinatorial characterization of resolution width. *J. Comput. System Sci.*, 74(3):323–334, 2008. Preliminary version in CCC'03. [doi:10.1016/j.jcss.2007.06.025] 476, 550

[6] SVEN BAUMER, JUAN LUIS ESTEBAN, AND JACOBO TORÁN: Minimally unsatisfiable CNF formulas. *Bulletin of the European Association for Theoretical Computer Science*, 74:190–192, 2001. 503

[7] ROBERTO J. BAYARDO, JR. AND ROBERT SCHRAG: Using CSP look-back techniques to solve real-world SAT instances. In *Proc. 14th Nat. Conf. on Artificial Intelligence (AAAI'97)*, pp. 203–208, 1997. CiteSeerX. [ACM:1867406.1867438] 474

[8] PAUL BEAME: Proof complexity. In STEVEN RUDICH AND AVI WIGDERSON, editors, *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*, pp. 199–246. American Mathematical Society, 2004. Available at author's home page. 475

[9] PAUL BEAME, CHRISTOPHER BECK, AND RUSSELL IMPAGLIAZZO: Time-space tradeoffs in resolution: superpolynomial lower bounds for superlinear space. In *Proc. 44th STOC*, pp. 213–232. ACM Press, 2012. [doi:10.1145/2213977.2213999] 550, 551

[10] PAUL BEAME, RICHARD M. KARP, TONIANN PITASSI, AND MICHAEL E. SAKS: The efficiency of resolution and Davis–Putnam procedures. *SIAM J. Comput.*, 31(4):1048–1075, 2002. Preliminary versions in FOCS'96 and STOC'98. [doi:10.1137/S0097539700369156] 475

[11] PAUL BEAME AND TONIANN PITASSI: Propositional proof complexity: Past, present and future. *Bulletin of the European Association for Theoretical Computer Science*, 65:66–89, 1998. Available at author's home page. See also updated version at ECCC. 475

[12] ELI BEN-SASSON: Size-space tradeoffs for resolution. *SIAM J. Comput.*, 38(6):2511–2525, 2009. Preliminary version in STOC'02. [doi:10.1137/080723880] 476, 477, 486, 503, 548

[13] ELI BEN-SASSON AND NICOLA GALESI: Space complexity of random formulae in resolution. *Random Structures & Algorithms*, 23(1):92–109, 2003. Preliminary version in CCC'01. [doi:10.1002/rsa.10089] 476

[14] ELI BEN-SASSON, RUSSELL IMPAGLIAZZO, AND AVI WIGDERSON: Near optimal separation of tree-like and general resolution. *Combinatorica*, 24(4):585–603, 2004. [doi:10.1007/s00493-004-0036-5] 476, 479, 486

[15] ELI BEN-SASSON AND JAKOB NORDSTRÖM: Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proc. 49th FOCS*, pp. 709–718. IEEE Comp. Soc. Press, 2008. [doi:10.1109/FOCS.2008.42] 478, 484, 488, 548, 549

[16] ELI BEN-SASSON AND JAKOB NORDSTRÖM: Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proc. 2nd Symp. Innovations in Computer Science (ICS'11)*, pp. 401–416, 2011. ICS'11. 549, 551

[17] ELI BEN-SASSON AND AVI WIGDERSON: Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001. Preliminary version in STOC'99. [doi:10.1145/375827.375835] 475, 476, 487, 549, 550

[18] ARCHIE BLAKE: *Canonical Expressions in Boolean Algebra*. Ph. D. thesis, University of Chicago, 1937. 475

[19] MARÍA LUISA BONET, JUAN LUIS ESTEBAN, NICOLA GALESI, AND JAN JOHANNSEN: On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM J. Comput.*, 30(5):1462–1484, 2000. Preliminary version in FOCS'98. [doi:10.1137/S0097539799352474] 486, 487

[20] MARÍA LUISA BONET AND NICOLA GALESI: Optimality of size-width tradeoffs for resolution. *Comput. Complexity*, 10(4):261–276, 2001. Preliminary version in FOCS'99. [doi:10.1007/s000370100000] 477

[21] JOSHUA BURESH-OPPENHEIM AND TONIANN PITASSI: The complexity of resolution refinements. *J. Symbolic Logic*, 72(4):1336–1352, 2007. Preliminary version in LICS'03. [doi:10.2178/jsl/1203350790] 486

[22] SAMUEL R. BUSS AND GYÖRGY TURÁN: Resolution proofs of generalized pigeonhole principles. *Theoret. Comput. Sci.*, 62(3):311–317, 1988. [doi:10.1016/0304-3975(88)90072-2] 475

[23] VAŠEK CHVÁTAL AND ENDRE SZEMERÉDI: Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988. [doi:10.1145/48014.48016] 475, 503

[24] STEPHEN A. COOK: The complexity of theorem-proving procedures. In *Proc. 3rd STOC*, pp. 151–158. ACM Press, 1971. [doi:10.1145/800157.805047] 474

[25] STEPHEN A. COOK: An observation on time-storage trade off. *J. Comput. System Sci.*, 9(3):308–316, 1974. Preliminary version in STOC'73. [doi:10.1016/S0022-0000(74)80046-2] 510

[26] STEPHEN A. COOK AND ROBERT A. RECKHOW: The relative efficiency of propositional proof systems. *J. Symbolic Logic*, 44(1):36–50, 1979. JSTOR. 474

[27] STEPHEN A. COOK AND RAVI SETHI: Storage requirements for deterministic polynomial time recognizable languages. *J. Comput. System Sci.*, 13(1):25–37, 1976. Preliminary version in STOC'74. [doi:10.1016/S0022-0000(76)80048-7] 480, 486, 508

[28] MARTIN DAVIS, GEORGE LOGEMANN, AND DONALD W. LOVELAND: A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962. [doi:10.1145/368273.368557] 474, 475

[29] MARTIN DAVIS AND HILARY PUTNAM: A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960. [doi:10.1145/321033.321034] 474, 475

[30] JUAN LUIS ESTEBAN, NICOLA GALESI, AND JOCHEN MESSNER: On the complexity of resolution with bounded conjunctions. *Theoret. Comput. Sci.*, 321(2-3):347–370, 2004. Preliminary version in ICALP'02. [doi:10.1016/j.tcs.2004.04.004] 476

[31] JUAN LUIS ESTEBAN AND JACOBO TORÁN: Space bounds for resolution. *Inform. and Comput.*, 171(1):84–97, 2001. Preliminary versions in STACS'99 and CSL'99. [doi:10.1006/inco.2001.2921] 476, 477, 485, 486, 550

[32] JUAN LUIS ESTEBAN AND JACOBO TORÁN: A combinatorial characterization of treelike resolution space. *Inform. Process. Lett.*, 87(6):295–300, 2003. [doi:10.1016/S0020-0190(03)00345-4] 476, 477, 548

[33] ZVI GALIL: On resolution with clauses of bounded size. *SIAM J. Comput.*, 6(3):444–459, 1977. Preliminary version found in STOC'75. [doi:10.1137/0206031] 475

[34] JOHN R. GILBERT, THOMAS LENGAUER, AND ROBERT ENDRE TARJAN: The pebbling problem is complete in polynomial space. *SIAM J. Comput.*, 9(3):513–524, 1980. Preliminary version in STOC'79. [doi:10.1137/0209038] 549

[35] JOHN R. GILBERT AND ROBERT ENDRE TARJAN: Variations of a pebble game on graphs. Technical Report STAN-CS-78-661, Stanford University, 1978. InfoLab. [ACM:892174] 508, 548, 549

[36] ARMIN HAKEN: The intractability of resolution. *Theoret. Comput. Sci.*, 39:297–308, 1985. [doi:10.1016/0304-3975(85)90144-6] 475

[37] ALEXANDER HERTEL: *Applications of Games to Propositional Proof Complexity*. Ph. D. thesis, University of Toronto, 2008. Available at author's home page. [ACM:1925663] 485

[38] PHILIPP HERTEL AND TONIANN PITASSI: The PSPACE-completeness of black-white pebbling. *SIAM J. Comput.*, 39(6):2622–2682, 2010. Preliminary version in FOCS'07. [doi:10.1137/080713513] 549

[39] JOHN E. HOPCROFT, WOLFGANG J. PAUL, AND LESLIE G. VALIANT: On time versus space. *J. ACM*, 24(2):332–337, 1977. Preliminary version in FOCS'75. [doi:10.1145/322003.322015] 477, 486

[40] MATTI JÄRVISALO, ARIE MATSLIAH, JAKOB NORDSTRÖM, AND STANISLAV ŽIVNÝ: Relating proof complexity measures and practical hardness of SAT. In *Proc. 18th Internat. Conf. on Principles and Practice of Constraint Programming (CP'12)*, pp. 316–331. Springer, 2012. [doi:10.1007/978-3-642-33558-7_25] 550

[41] HENRY A. KAUTZ AND BART SELMAN: The state of SAT. *Discr. Appl. Math.*, 155(12):1514–1524, 2007. [doi:10.1016/j.dam.2006.10.004] 474

[42] MARIA M. KLAWE: A tight bound for black and white pebbles on the pyramid. *J. ACM*, 32(1):218–228, 1985. Preliminary version in FOCS'83. [doi:10.1145/2455.214115] 483, 484, 508, 511, 512, 513, 514, 516, 517, 520, 522, 523, 524, 548

[43] OLIVER KULLMANN: An application of matroid theory to the SAT problem. In *Proc. 15th IEEE Conf. on Computational Complexity (CCC'00)*, pp. 116–124. IEEE Comp. Soc. Press, 2000. [doi:10.1109/CCC.2000.856741] 503

[44] THOMAS LENGAUER AND ROBERT ENDRE TARJAN: The space complexity of pebble games on trees. *Inform. Process. Lett.*, 10(4-5):184–188, 1980. [doi:10.1016/0020-0190(80)90136-2] 508, 510

[45] JOÃO P. MARQUES-SILVA: Practical applications of Boolean satisfiability. In *9th Internat. Workshop on Discrete Event Systems (WODES'08)*, pp. 74–80, 2008. [doi:10.1109/WODES.2008.4605925] 474

[46] JOÃO P. MARQUES-SILVA AND KAREM A. SAKALLAH: GRASP: A search algorithm for propositional satisfiability. *IEEE Trans. Computers*, 48(5):506–521, 1999. Preliminary version in ICCAD'96. [doi:10.1109/12.769433] 474

[47] JAKOB NORDSTRÖM: Narrow proofs may be spacious: Separating space and width in resolution. *SIAM J. Comput.*, 39(1):59–121, 2009. Preliminary version in STOC'06. [doi:10.1137/060668250] 476, 477, 484, 508, 548

[48] JAKOB NORDSTRÖM: New wine into old wineskins: A survey of some pebbling classics with supplemental results. Manuscript in preparation. To appear in *Foundations and Trends in Theoretical Computer Science*. Current draft version available at author's home page, 2013. 486, 516, 524

[49] JAKOB NORDSTRÖM: Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 2013. To appear. Available at author's home page. 484

[50] JAKOB NORDSTRÖM AND JOHAN HÅSTAD: Towards an optimal separation of space and length in resolution. In *Proc. 40th STOC*, pp. 701–710. ACM Press, 2008. [doi:10.1145/1374376.1374478] 488, 527

[51] 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. 486

[52] RAN RAZ: Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004. Preliminary version in STOC'02. [doi:10.1145/972639.972640] 475

[53] RAN RAZ AND PIERRE MCKENZIE: Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999. Preliminary version in FOCS'97. [doi:10.1007/s004930050062] 487

[54] ALEXANDER A. RAZBOROV: Resolution lower bounds for the weak functional pigeonhole principle. *Theoret. Comput. Sci.*, 303(1):233–243, 2003. [doi:10.1016/S0304-3975(02)00453-X] 475

[55] ALEXANDER A. RAZBOROV: Resolution lower bounds for perfect matching principles. *J. Comput. System Sci.*, 69(1):3–27, 2004. Preliminary version in CCC'02. [doi:10.1016/j.jcss.2004.01.004] 475

[56] JOHN ALAN ROBINSON: A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965. [doi:10.1145/321250.321253] 475

[57] ASHISH SABHARWAL, PAUL BEAME, AND HENRY A. KAUTZ: Using problem structure for efficient clause learning. In *6th Internat. Conf. on Theory and Applications of Satisfiability Testing (SAT'03)*, pp. 242–256, 2004. [doi:10.1007/978-3-540-24605-3_19] 550

[58] The international SAT Competitions web page. `http://www.satcompetition.org`. 474

[59] NATHAN SEGERLIND: The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007. Bulletin of Symbolic Logic. 475, 548

[60] JACOBO TORÁN: Space and width in propositional resolution. *Bulletin of the EATCS*, 83:86–104, 2004. Universität Ulm. 477, 548

[61] GRIGORI TSEITIN: On the complexity of derivation in propositional calculus. In A. O. SILENKO, editor, *Structures in Constructive Mathematics and mathematical Logic, Part II*, pp. 115–125. Consultants Bureau, New York-London, 1968. 475

[62] ALASDAIR URQUHART: Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987. [doi:10.1145/7531.8928] 475

AUTHORS

Jakob Nordström
Assistant Professor
KTH Royal Institute of Technology, Stockholm, Sweden jakobn@kth.se
http://www.csc.kth.se/~jakobn

Johan Håstad
Professor
KTH Royal Institute of Technology, Stockholm, Sweden johanh@kth.se
http://www.csc.kth.se/~johanh

ABOUT THE AUTHORS

JAKOB NORDSTRÖM received his Master of Science in Computer Science and Mathematics from Stockholm University in 2001, and his Ph. D. in Computer Science from KTH Royal Institute of Technology in 2008, where he was advised by Johan Håstad. During the academic years 2008-09 and 2009-10 he was a postdoctoral researcher at the Massachusetts Institute of Technology (MIT). Since 2011 he has been an Assistant Professor at the School of Computer Science and Communication at KTH. So far, he has worked mostly on proof complexity and on questions at the intersection of proof complexity and SAT solving.

In 1997-1998, Jakob served as a military interpreter at the Swedish Armed Forces Language Institute (Försvarets tolkskola). In parallel with his studies and later his research, he has worked as a Russian interpreter, engaged among others for His Majesty the King of Sweden and the Swedish Prime Minister. He also has a Diploma in Choir Conducting with extended Music Theory from the Tallinn Music Upper Secondary School, Estonia. During the period 1994-1999, he was the artistic director of Collegium Vocale Stockholm, a vocal ensemble performing mainly Renaissance and Baroque music.

JOHAN HÅSTAD received his Bachelor of Science from Stockholm University in 1981, his Master of Science from Uppsala University in 1984 and his Ph. D. from MIT in 1986 under the supervision of Shafi Goldwasser. Johan was appointed Associate Professor at KTH Royal Institute of Technology in Stockholm, Sweden, in 1988 and was promoted to Professor in 1992. He has been a member of the Swedish Royal Academy of Sciences since 2001. He has research interests within several subareas of Theory of Algorithms and Complexity theory but has, in recent years, mainly focused on the inapproximability of NP-hard optimization problems.