Electronic Colloquium on Computational Complexity, Report No. 26 (2008)

Towards an Optimal Separation of Space and Length in Resolution*

Jakob Nordström† Johan Håstad

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

February 29, 2008

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

In this paper, we present some evidence that the true answer should be that the latter case holds and provide a possible roadmap for how such an optimal separation result could be obtained. We do this 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 in (Nordström 2006) from logarithmic to polynomial.

Also, continuing the line of research initiated by (Ben-Sasson 2002) into trade-offs between different proof complexity measures, we present a simplified proof of the recent length-space trade-off result in (Hertel and Pitassi 2007), and show how our ideas can be used to prove a couple of other exponential trade-offs in resolution.

1 Introduction

Ever since the fundamental NP-completeness result of Cook [21], 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

*This is the full-length version of the paper [44] to appear at *STOC '08*.

†Research supported in part by grants from the foundations *Johan och Jakob Söderbergs stiftelse* and *Sven och Dagmar Saléns stiftelse*.

ISSN 1433-8092

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 somewhat surprising aspect of this development is that the most successful SAT-solvers to date are still variants of the resolution-based Davis-Putnam-Logemann-Loveland (DPLL) procedure [25, 26] augmented with *clause learning*. For instance, the great majority of the best algorithms at the 2007 round of the international SAT competitions [53] fit this description. DPLL procedures perform a recursive backtrack search in the space of partial truth value assignments. The idea behind clause learning, or *conflict-driven learning*, is that at each failure (backtrack) point in the search tree, the system derives a reason for the inconsistency in the form of a new clause and then adds this clause to the original CNF formula ("learning" the clause). This can save a lot of work later on in the proof search, when some other partial truth value assignment fails for similar reasons. The main bottleneck for this approach, other than the obvious one of time, is the amount of memory used by the algorithms. Since there is only a finite 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. [9, 36, 51] for a more detailed discussion of clause learning (and SAT-solving in general) with examples of applications.

The study of proof complexity originated with the seminal paper of Cook and Reckhow [23]. 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 fundamental 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 [23] that $\mathsf{NP} = \mathsf{co\text{-}NP}$ if and only if there exists a polynomially bounded propositional proof system. Thus, if it could be shown that there are no polynomially bounded proof systems for propositional tautologies, $\mathsf{P} \neq \mathsf{NP}$ would follow as 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 a very 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, 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 [7, 10, 54].

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 [16] and began to be investigated in connection with automated theorem proving in the 1960s [25, 26, 50]. 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 most straightforward to prove bounds on the *length* of refutations, i.e., the number of clauses, rather than on the total size of refutations. The length and size measures are easily seen to be polynomially related. In 1968, Tseitin [58] 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 [32] proved the first superpolynomial lower bound for general resolution. This weakly exponential bound of Haken has later been followed by many other strong results, among others truly exponential lower bound on resolution refutation length for different formula families in, for instance, [8, 15, 20, 59].

A second complexity measure for resolution, first made explicit by Galil [30], is the *width*, measured as the maximal size of a clause in the refutation. Ben-Sasson and Wigderson [15] showed that the minimal width $W(F \vdash 0)$ of any resolution refutation of a k-CNF formula F is bounded from above by the minimal refutation length $L(F \vdash 0)$ by

$$W(F \vdash 0) = \mathrm{O}\Bigl(\sqrt{n \log L(F \vdash 0)}\Bigr) \ , \tag{1.1}$$

where n is the number of variables in F. Since it is also easy to see that resolution refutations of polynomial-size formulas in small width must necessarily be short (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 [15] 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* in resolution was initiated by Esteban and Torán [28, 56]. Intuitively, the space $Sp(\pi)$ of a resolution refutation π is the maximal number of clauses one needs to keep in memory while verifying the refutation, and the space $Sp(F \vdash 0)$ of refuting F is defined as the minimal space of any 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, 27, 29]. 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 resolution 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

$$W(F \vdash 0) \leq Sp(F \vdash 0) + \mathrm{O}(1) \tag{1.2}$$

holds for any k-CNF formula F, where the (small) constant term depends on k. In [42], 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 $W(F_n \vdash 0) = \mathrm{O}(1)$ but requiring space $Sp(F_n \vdash 0) = \Theta(\log n)$.

The space measure discussed above is known as *clause space*. A less well-studied space measure, introduced by Alekhnovich et al. [2], is *variable space*, which counts the maximal number of variable occurrences that must be kept in memory simultaneously. Ben-Sasson [11] used this measure to obtain a trade-off result for clause space versus width in resolution, proving that there are k-CNF formulas F_n that can be refuted in constant clause space and constant width, but for which any refutation π_n must have $Sp(\pi_n) \cdot W(\pi_n) = \Omega(n/\log n)$. More recently, Hertel and Pitassi [33] showed that there are CNF formulas F_n for which any refutation of F_n in minimal variable space $VarSp(F_n \vdash 0)$ must have exponential length, but by adding just 3 extra units of storage one can instead get a resolution refutation in linear length.

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 still remain 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.

Equation (1.1) says that short refutation length implies narrow refutation width. Combining Equation (1.2) with the observation above that narrow refutations are trivially short, we get a similar statement that small refutation clause space implies short refutation length. Note, however, that this does *not* mean that there is a refutation that is both short and narrow, or that any small-space refutation must also be short. The reason is that the resolution refutations on the left- and right-hand sides of (1.1) and (1.2) need not (and in general will not) be the same one.

In view of the minimum-width proof search heuristic mentioned above, an important question is whether short refutation length of a formula does in fact entail that there is a refutation of it that is both short and narrow. Also, it would be interesting to know if small space of a refutation implies that it is short. It is not known whether there are such connections or whether on the contrary there exist some kind of trade-off phenomena here similar to the one for space and width in [11].

A third, even more interesting problem is to clarify the relation between length and clause space. For width, rewriting the bound in (1.1) in terms of the number of clauses $|F_n|$ instead of the number of variables we get that 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 [18] shows that there is a k-CNF formula family $\{F_n\}_{n=1}^{\infty}$ with $W(F_n \vdash 0) = \Omega\big(\sqrt[3]{|F_n|}\big)$ but $L(F_n \vdash 0) = \mathrm{O}(|F_n|)$. 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 space and length 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 [28, 34] that $Sp(\pi) = \mathrm{O}(L/\log L)$. This gives an upper bound on any possible separation of the two measures. But is there a Ben-Sasson–Wigderson kind of 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 $Sp(F_n \vdash 0) = \Omega\big(L(F_n \vdash 0)/\log L(F_n \vdash 0)\big)$ in terms of length?

We note that for the restricted case of so-called tree-like resolution, [28] 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, for instance, [11, 29, 57], but there seems to have 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 some evidence 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 pyramid graphs.

Theorem 1.1. *The clause space of refuting pebbling contradictions over pyramids 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 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 [42].

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 $L(F_n \vdash 0) = \mathrm{O}(n)$ and width $W(F_n \vdash 0) = \mathrm{O}(1)$ but require clause space $Sp(F_n \vdash 0) = \Theta(\sqrt{n})$.*

In addition to our main result, we also make the the observation that the proof of the recent trade-off result in [33] can be greatly simplified, and the parameters slightly improved. Using similar ideas, we can also prove exponential trade-offs for length with respect to clause space and width. Namely, we show that there are k-CNF formulas such that if we insist on finding the resolution refutation in smallest clause space or smallest width, respectively, then we have to pay with an exponential increase in length. We state the theorem only for length versus clause space.

Theorem 1.3. *There is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

- *The minimal clause space of refuting F_n in resolution is $Sp(F_n \vdash 0) = \Theta\big(\sqrt[3]{n}\big)$.*
- *Any resolution refutation $\pi : F_n \vdash 0$ in minimal clause space must have length $L(\pi) = \exp\big(\Omega\big(\sqrt[3]{n}\big)\big)$.*
- *There are resolution refutations $\pi' : F_n \vdash 0$ in asymptotically minimal clause space $Sp(\pi') = \mathrm{O}\big(Sp(F_n \vdash 0)\big)$ and length $L(\pi') = \mathrm{O}(n)$, i.e., linear in the formula size.*

A theorem of exactly the same form can be proven for length versus width as well.

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.[1] The refutation ends when we reach the contradictory empty clause. The *length* of a resolution refutation is the number of distinct 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 $L(F \vdash 0)$, $W(F \vdash 0)$ and $Sp(F \vdash 0)$ 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. 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 the empty graph G 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 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 1 for a small example. Very briefly, pebbling contradictions are constructed as follows:

- Associate d variables $x(v)_1, \ldots, x(v)_d$ with each vertex v (in Figure 1 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 1.
- 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 1 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, 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.

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* [24] modeling the non-deterministic computations described by the underlying graphs on the

[1] 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 that are implied by the set of clauses currently on the blackboard.

$$
\begin{aligned}
& (x(r)_1 \lor x(r)_2) \\
\land\ & (x(s)_1 \lor x(s)_2) \\
\land\ & (x(t)_1 \lor x(t)_2) \\
\land\ & (\overline{x(r)}_1 \lor \overline{x(s)}_1 \lor x(u)_1 \lor x(u)_2) \\
\land\ & (\overline{x(r)}_1 \lor \overline{x(s)}_2 \lor x(u)_1 \lor x(u)_2) \\
\land\ & (\overline{x(r)}_2 \lor \overline{x(s)}_1 \lor x(u)_1 \lor x(u)_2) \\
\land\ & (\overline{x(r)}_2 \lor \overline{x(s)}_2 \lor x(u)_1 \lor x(u)_2) \\
\land\ & (\overline{x(s)}_1 \lor \overline{x(t)}_1 \lor x(v)_1 \lor x(v)_2) \\
\land\ & (\overline{x(s)}_1 \lor \overline{x(t)}_2 \lor x(v)_1 \lor x(v)_2) \\
\land\ & (\overline{x(s)}_2 \lor \overline{x(t)}_1 \lor x(v)_1 \lor x(v)_2) \\
\land\ & (\overline{x(s)}_2 \lor \overline{x(t)}_2 \lor x(v)_1 \lor x(v)_2)
\end{aligned}
\qquad
\begin{aligned}
\land\ & (\overline{x(u)}_1 \lor \overline{x(v)}_1 \lor x(z)_1 \lor x(z)_2) \\
\land\ & (\overline{x(u)}_1 \lor \overline{x(v)}_2 \lor x(z)_1 \lor x(z)_2) \\
\land\ & (\overline{x(u)}_2 \lor \overline{x(v)}_1 \lor x(z)_1 \lor x(z)_2) \\
\land\ & (\overline{x(u)}_2 \lor \overline{x(v)}_2 \lor x(z)_1 \lor x(z)_2) \\
\land\ & \overline{x(z)}_1 \\
\land\ & \overline{x(z)}_2
\end{aligned}
$$

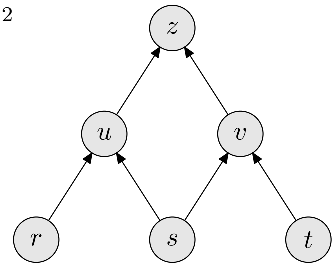


Figure 1: The pebbling contradiction $Peb^2_{\Pi_2}$ for the pyramid graph Π_2 of height 2.

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.

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

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

2. Then, prove that this interpretation of clauses in terms of pebbles 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 captures clause space in the sense that if the content of the blackboard induces N pebbles on the graph, then there must be at least N clauses on the blackboard.

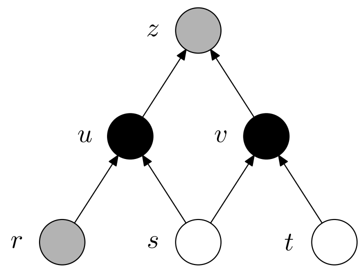
Combining the above with known lower bounds on the pebbling price of G, this would imply a lower bound on the refutation space of pebbling contradictions and a separation from length and width. For clarity, let us spell out what the formal argument of this would look like.

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, this idea does not quite 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.

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

(a) Clauses on blackboard.



(b) Corresponding pebbles in the graph.

Figure 2: Example of intuitive correspondence between sets of clauses and pebbles.

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 these assumptions 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 in 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 the black pebbles above w in the graph. Needing to assume the truth of w is the opposite of knowing the truth of w, so extending the reasoning above we get 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.

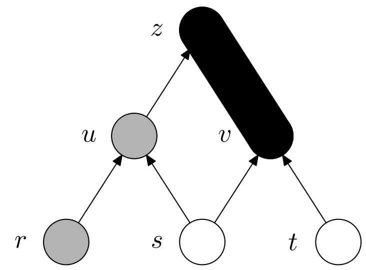
Using this intuitive correspondence, we can translate sets of clauses in a resolution refutation of Peb_G^d into black and white pebbles in G as in Figure 2. It is easy to see that if we assume $x(s)_1 \vee x(s)_2$ and $x(t)_1 \vee x(t)_2$, this assumption together with the clauses on the blackboard in Figure 2(a) imply $x(v)_1 \vee x(v)_2$, so v should be black-pebbled and s and t white-pebbled in Figure 2(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 arguably quite straightforward, and seems to yield well-behaved black-white pebblings for all "sensible" resolution refutations of Peb_G^d.

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 2(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 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 3(a), will then be translated into white

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

(a) New set of clauses on blackboard.



(b) Corresponding blobs and pebbles.

Figure 3: Intepreting sets of clauses as black blobs and white pebbles.

pebbles on s and t as before and a black blob covering both v and z in Figure 3(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, 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 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 $\mathsf{cost}(\mathcal{P}_\pi) \leq Sp(\pi) + \mathrm{O}(1)$.*

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 [37].

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 $d > 1$ defined over the pyramid graph of height h. Then the clause space of refuting $Peb^d_{\Pi_h}$ by resolution is $Sp(Peb^d_{\Pi_h} \vdash 0) = \Theta(h)$.*

Proof. The upper bound $Sp(Peb^d_{\Pi_h} \vdash 0) = O(h)$ is easy. A pyramid of height h can be pebbled with $h + O(1)$ 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 $\mathit{cost}(\mathcal{P}_\pi) = O(Sp(\pi))$ 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 of G (compare Figure 1). 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 $L(Peb^d_G \vdash 0) = O(n)$ and $W(Peb^d_G \vdash 0) = O(1)$. 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 $k \geq 4$, there is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $O(n)$ such that $L(F_n \vdash 0) = O(n)$ and $W(F_n \vdash 0) = O(1)$ but $Sp(F_n \vdash 0) = \Theta(\sqrt{n})$.*

2.4 Overview of Trade-off Results

Let us also quickly sketch the ideas (or tricks, really) used to prove our trade-off theorems for resolution.

We show the following version of the length-variable space trade-off theorem of Hertel and Pitassi [33], with somewhat improved parameters and a very much simpler proof.

Theorem 2.4. *There is a family of CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

- *The minimal variable space of refuting F_n in resolution is $VarSp(F_n \vdash 0) = \Theta(n)$.*
- *Any resolution refutation $\pi : F_n \vdash 0$ in minimal variable space has length $\exp(\Omega(\sqrt{n}))$.*
- *Adding at most 2 extra units of storage, it is possible to obtain a resolution refutation π' in variable space $VarSp(\pi') = VarSp(F_n \vdash 0) + 3 = \Theta(n)$ and length $L(\pi') = O(n)$, i.e., linear in the formula size.*

The idea behind our proof is as follows. Take formulas G_n that are really hard for resolution and formulas H_m which have short refutations but require linear variable space, and set $F_n = G_n \wedge H_m$ for m chosen so that $VarSp\big(H_m \vdash 0\big)$ is only just larger than $VarSp\big(G_n \vdash 0\big)$. Then refutations in minimal variable space will have to take care of G_n, which requires exponential length, but adding one or two literals to the memory we can attack H_m instead in linear length.

The trade-off result in Theorem 1.3 for length versus clause space and its twin theorem for length versus width are shown using similar ideas.

2.5 Paper Organization

Section 3 provides formal definitions of the concepts introduced in Sections 1 and 2, and Section 4 gives precise statements of the results mentioned there, as well as some other result relevant to this paper. The easy proofs of our trade-off theorems are then immediately presented in Section 5.

The bulk of the paper is spent proving our main result in Theorem 1.1. In Section 6, we define our modified pebble game, the "blob-pebble game", that we will use to analyze resolution refutations of pebbling contradictions. In Section 7 we prove that resolution refutations can be translated into pebblings in this game, which is Theorem 2.1 in Section 2.3. In Section 8, 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 detailed description of the lower bound on black-white pebbling of [37] in Section 9 (with a somewhat simplified proof that might be of independent interest), in Section 10 we generalize this result in a nontrivial way 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 11 by giving suggestions for further research.

3 Formal Preliminaries

In this section, we define resolution, pebble games and pebbling contradictions.

3.1 The Resolution Proof System

A *literal* is either a propositional logic variable or its negation, denoted x and $\overline{x}$, respectively. 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, all clauses C are assumed to be nontrivial in the sense that all literals in C are pairwise strictly distinct (otherwise C is trivially true). We say that C is a *subclause* of D if $C \subseteq D$. A clause containing at most k literals is called a k-*clause*.

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

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

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

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

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

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

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 0 (the clause with no literals) from F. Perhaps somewhat confusingly, this is sometimes also referred to as a *resolution proof* of F.

For a formula F and a set of formulas $\mathcal{G} = \{G_1, \ldots, G_n\}$, we say that $\mathcal{G}$ *implies* F, denoted $\mathcal{G} \vDash F$, if every truth value assignment satisfying all formulas $G \in \mathcal{G}$ satisfies F as well. It is

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

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. We define the length of deriving a clause A from a formula F as $L(F \vdash A) = \min_{\pi : F \vdash A}\{L(\pi)\}$, where the minimum is taken over all resolution derivations of A. In particular, the length of refuting F by resolution is denoted $L(F \vdash 0)$. The length of refuting F by tree-like resolution $L_{\mathfrak{T}}(F \vdash 0)$ is defined by taking the minimum over all tree-like resolution refutations π_T of F.

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

We next define the measure of *space*. Following the exposition in [28], 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 *variable space* 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[3] of the following rules:

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

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

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

A resolution derivation $\pi : F \vdash A$ of a clause A from a formula F is a derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ such that $\mathbb{C}_\tau = \{A\}$. A *resolution refutation* of F is a derivation of the empty clause 0 from F.

Definition 3.2 (Clause space [2, 11]). The *clause space* of a resolution derivation $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is $\max_{t \in [\tau]}\{|\mathbb{C}_t|\}$. The clause space of deriving A from F is $Sp(F \vdash A) = \min_{\pi : F \vdash A}\{Sp(\pi)\}$, and $Sp(F \vdash 0)$ denotes the minimum clause space of any resolution refutation of F.

Definition 3.3 (Variable space [2]). The *variable space* of a configuration $\mathbb{C}$ is $VarSp(\mathbb{C}) = \sum_{C \in \mathbb{C}} W(C)$. The variable space of a derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is $\max_{t \in [\tau]}\{VarSp(\mathbb{C}_t)\}$, and $VarSp(F \vdash 0)$ is the minimum variable space of any resolution refutation of F.

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

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

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

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

3.2 Pebble Games and Pebbling Contradictions

Pebble games were devised for studying programming languages and compiler construction, but have found a variety of applications in computational complexity theory. In connection with resolution, pebble games have been employed both to analyze resolution derivations with respect to how much memory they consume (using the original definition of space in [28]) and to construct CNF formulas which are hard for different variants of resolution in various respects (see for example [3, 14, 17, 19]). An excellent survey of pebbling up to ca 1980 is [48].

The black pebbling price of a DAG G captures the memory space, i.e., the number of registers, required to perform the deterministic computation described by G. The space of a non-deterministic computation is measured by the black-white pebbling price of G. We say that vertices of G with indegree 0 are *sources* and that vertices with outdegree 0 are *sinks* or *targets*. In the following, unless otherwise stated we will assume that all DAGs under discussion have a unique sink and this sink will always be denoted z. The next definition is adapted from [24], though we use the established pebbling terminology introduced by [34].

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

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

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

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

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

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

The *cost* of a pebble configuration $\mathbb{P} = (B, W)$ is $\mathit{cost}(\mathbb{P}) = |B \cup W|$ and the cost of a pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is $\max_{0 \le t \le \tau}\{\mathit{cost}(\mathbb{P}_t)\}$. 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 $\mathsf{Peb}(G)$, 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 target 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 [17, 49].

Definition 3.5 (Pebbling contradiction [15]). Suppose that G is a DAG with sources S, a unique target 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*),
- $\overline{x(z)}_i$ for all $i \in [d]$ (*target 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*).

The formula Peb^d_G is a (2+d)-CNF formula with $\mathrm{O}\bigl(d^2 \cdot |V(G)|\bigr)$ clauses over $d \cdot |V(G)|$ variables. An example pebbling contradiction is presented in Figure 1 on page 7.

4 Review of Related Work

This section is an overview of related work, including formal statements of some previously known results that we will need. At the end of Section 4.3 we also try to provide some of the intuition behind the result proven in this paper.

4.1 General Results About Resolution

It is not hard to show that any CNF formula F over n variables is refutable in length $2^{n+1} - 1$ and width n. Esteban and Torán [28] proved that the clause space of refuting F is upper-bounded by the formula size. More precisely, the minimal clause space is at most the number of clauses, or the number of variables, plus a small constant, or in formal notation $Sp(F \vdash 0) \leq \min\bigl\{|F|, |\mathit{Vars}(F)|\bigr\} + \mathrm{O}(1)$.

We will need the fact that there are polynomial-size families of k-CNF formulas that are very hard with respect to length, width and clause space, essentially meeting the upper bounds just stated.

Theorem 4.1 ([2, 8, 13, 15, 20, 56, 59]). *There are arbitrarily large unsatisfiable 3-CNF formulas F_n of size $\Theta(n)$ with $\Theta(n)$ clauses and $\Theta(n)$ variables for which it holds that $L(F_n \vdash 0) = \exp(\Theta(n))$, $W(F_n \vdash 0) = \Theta(n)$ and $Sp(F_n \vdash 0) = \Theta(n)$.*

Clearly, for such formulas F_n it must also hold that $\Omega(n) = \mathit{VarSp}(F_n \vdash 0) = \mathrm{O}(n^2)$. We note in passing that determining the exact variable space complexity of a formula family as in Theorem 4.1 was mentioned as an open problem in [2]. To the best of our knowledge this problem is still unsolved.

If a resolution refutation has constant width, it is easy to see that it must be of size polynomial in the number of variables (just count the maximum possible number of distinct clauses). Conversely, if all refutations of a formula are very wide, it seems reasonable that any refutation of this formula must be very long as well. This intuition was made precise by Ben-Sasson and Wigderson [15]. We state their theorem in the more explicit form of Segerlind [54].

Theorem 4.2 ([15]). *The width of refuting a CNF formula F is bounded from above by*

$$W(F \vdash 0) \leq W(F) + 1 + 3\sqrt{n \ln L(F \vdash 0)} \enspace ,$$

where n is the number of variables in F.

Bonet and Galesi [18] showed that this bound on width in terms of length is essentially optimal. For the special case of tree-like resolution, however, it is possible get rid of the dependence of the number of variables and obtain a tighter bound.

Theorem 4.3 ([15]). *The width of refuting a CNF formula F in tree-like resolution is bounded from above by $W(F \vdash 0) \leq W(F) + \log L_{\mathfrak{T}}(F \vdash 0)$.*

For reference, we collect the result in [18] together with some other bounds showing that there are formulas that are easy with respect to length but moderately hard with respect to width and clause space and state them as a theorem.[4]

Theorem 4.4 ([2, 18, 55]). *There are arbitrarily large unsatisfiable 3-CNF formulas F_n of size $\Theta(n^3)$ with $\Theta(n^3)$ clauses and $\Theta(n^2)$ variables such that $W(F_n \vdash 0) = \Theta(n)$ and $\mathit{Sp}(F_n \vdash 0) = \Theta(n)$, but for which there are resolution refutations $\pi_n : F_n \vdash 0$ in length $L(\pi_n) = \mathrm{O}(n^3)$, width $W(\pi_n) = \mathrm{O}(n)$ and clause space $\mathit{Sp}(\pi_n) = \mathrm{O}(n)$.*

As was mentioned above, the fact that all known lower bounds on refutation clause space coincided with lower bounds on width lead to the conjecture that the width measure is a lower bound for the clause space measure. This conjecture was proven true by Atserias and Dalmau [5].

Theorem 4.5 ([5]). *For any CNF formula F, it holds that $\mathit{Sp}(F \vdash 0) - 3 \geq W(F \vdash 0) - W(F)$.*

In other words, the extra clause space exceeding the minimum 3 needed for any resolution derivation is bounded from below by the extra width exceeding the width of the formula. This inequality was later shown by the first author to be asymptotically strict in the following sense.

Theorem 4.6 ([42]). *For all $k \geq 4$, there is a family $\{F_n\}_{n=1}^{\infty}$ of k-CNF formulas of size $\Theta(n)$ such that $L(F_n \vdash 0) = \mathrm{O}(n)$ and $W(F_n \vdash 0) = \mathrm{O}(1)$ but $\mathit{Sp}(F_n \vdash 0) = \Theta(\log n)$.*

An immediate corollary of Theorem 4.5 is that for polynomial-size k-CNF formulas constant clause space implies polynomial proof length. We are interested in finding out what holds in the other direction, i.e., if upper bounds on length imply upper bounds on space.

For the special case of tree-like resolution, it is known that there is an upper bound on clause space in terms of length exactly analogous to the one on width in terms of length in Theorem 4.3.

[4] Note that [18], where an explicit resolution refutation upper-bounding the proof complexity measures is presented, does not talk about clause space, but it is straightforward to verify that the refutation there can be carried out in length $\mathrm{O}(n^3)$ and clause space $\mathrm{O}(n)$.

Theorem 4.7 ([28]). *For any tree-like resolution refutation π of a CNF formula F it holds that $Sp(\pi) \leq \lceil \log L(\pi) \rceil + 2$. In particular, $Sp(F \vdash 0) \leq \lceil \log L_{\mathfrak{T}}(F \vdash 0) \rceil + 2$.*

For general resolution, since clause space is lower-bounded by width according to Theorem 4.5, the separation of width and length of [18] in Theorem 4.4 tells us that k-CNF formulas refutable in polynomial length can still have "somewhat spacious" minimum-space refutations. But exactly how spacious can they be? Does space behave as width with respect to length also in general resolution, or can one get stronger lower bounds on space for formulas refutable in polynomial length?

All polynomial lower bounds on clause space known prior to this paper can be explained as immediate consequences of Theorem 4.5 applied on lower bounds on width. Clearly, any space lower bounds derived in this way cannot get us beyond the "Ben-Sasson–Wigderson barrier" implied by Theorem 4.2 saying that if the width of refuting F is $\omega\bigl(\sqrt{|F| \log |F|}\bigr)$, then the length of refuting F must be superpolynomial in $|F|$. Also, since matching upper bounds on clause space have been known for all of these formula families, they have not been candidates for showing stronger separations of space and length. Thus, the best known separation of clause space and length has been the formulas in Theorem 4.4 refutable in linear length $L(F_n \vdash 0) = \mathrm{O}(|F_n|)$ but requiring space $Sp(F_n \vdash 0) = \Theta\bigl(\sqrt[3]{|F_n|}\bigr)$, as implied by the same bound on width.

Let us also discuss upper bounds on what kind of separations are a priori possible. Given any resolution refutation $\pi : F \vdash 0$, we can write down its DAG representation G_π (described on page 12) with $L(\pi)$ vertices corresponding to the clauses, and with all non-source vertices having fan-in 2. We can then transform π into as space-efficient a refutation as possible by considering an optimal black pebbling of G_π as follows: when a pebble is placed on a vertex we derive the corresponding clause, and when the pebble is removed again we erase the clause from memory. This yields a refutation π' in clause space $\mathsf{Peb}(G_\pi)$ (incidentally, this is the original definition in [28] of the clause space of a resolution refution π). Since it is known that any constant indegree DAG on n vertices can be black-pebbled in cost $\mathrm{O}(n/\log n)$ (see Theorem 4.10), this shows that $Sp(F \vdash 0) = \mathrm{O}\bigl(L(F \vdash 0)/\log L(F \vdash 0)\bigr)$ is a trivial upper bound on space in terms of length.

Now we can rephrase the question above about space and length in the following way: Is there a Ben-Sasson–Wigderson kind of lower bound, say $L(F \vdash 0) = \exp\bigl(\Omega\bigl(Sp(F \vdash 0)^2/|F|\bigr)\bigr)$ or so, on length in terms of space? Or do there exist k-CNF formulas F with short refutations but maximum possible refutation space $Sp(F \vdash 0) = \Omega\bigl(L(F \vdash 0)/\log L(F \vdash 0)\bigr)$ in terms of length? Note that the refutation length $L(F \vdash 0)$ must indeed be short in this case—essentially linear, since any formula F can be refuted in space $\mathrm{O}(|F|)$ as was noted above. Or is the relation between refutation space and refutation length somewhere in between these extremes?

This is the main question addressed in this paper. We believe that clause space and length can be strongly separated in the sense that there are formula families with maximum possible refutation space in terms of length. As a step towards proving this we improve the lower bound in Theorem 4.6 from $\Theta(\log n)$ to $\Theta(\sqrt{n})$, thus providing the first polynomial lower bound on space that is not the consequence of a corresponding bound on width. We next review some results about the tools that we use to do this.

4.2 Results About Pebble Games

There is an extensive literature on pebbling, mostly from the 70s and 80s. We just quickly mention four results relevant to this paper.

Perhaps the simplest graphs to pebble are complete binary trees T_h of height h. The black pebbling price of T_h can be established by an easy induction over the tree height. For black-white pebbling, general bounds for the pebbling price of trees of any arity were presented in [39]. For the case of binary trees, this result can be simplified to an exact equality (a proof of which can be found in Section 4 of [41]).

Theorem 4.8. *For a complete binary tree T_h of height $h \geq 1$ it holds that $\textsf{Peb}(T_h) = h + 2$ and $\textsf{BW-Peb}(T_h) = \lfloor \frac{h}{2} \rfloor + 3$.*

In this paper, we will focus on pyramid graphs, an example of which can be found in Figure 1.

Theorem 4.9 ([22, 37]). *For a pyramid graph Π_h of height $h \geq 1$ it holds that $\textsf{Peb}(\Pi_h) = h + 2$ and $\textsf{BW-Peb}(\Pi_h) = h/2 + \mathrm{O}(1)$.*

As we wrote in Section 2, we are interested in DAGs with as high a pebbling price as possible measured in terms of the number of vertices. For a DAG G with n vertices and constant in-degree, the best we can hope for is $\mathrm{O}(n/\log n)$.

Theorem 4.10 ([34]). *For directed acyclic graphs G with n vertices and constant maximum indegree, it holds that $\textsf{Peb}(G) = \mathrm{O}\big(n/\log n\big)$.*

This bound is asymptotically tight both for black and black-white pebbling.

Theorem 4.11 ([31, 47]). *There is a family of explicitly constructible[5] DAGs G_n with $\Theta(n)$ vertices and vertex indegrees 0 or 2 such that $\textsf{Peb}(G) = \Theta(n/\log n)$ and $\textsf{BW-Peb}(G) = \Theta(n/\log n)$.*

It should be pointed out that although the black and black-white pebbling prices coincide asymptotically in all of the theorems above, this is not the case in general. In [35], a family of DAGs with a quadratic difference in the number of pebbles between the black and the black-white pebble game was presented. We note that this is the best separation possible, since by [40] the difference in black and black-white pebbling price can be at most quadratic.

4.3 Results About Pebbling Contradictions Plus Some Intuition

Although any constant indegree will be fine for the results covered in this subsection, we restrict our attention to DAGs with vertex indegrees 0 or 2 since these are the graphs that will be studied in the rest of this paper.

It was observed in [14] that Peb^d_G can be refuted in resolution by deriving $\bigvee_{i=1}^{d} x(v)_i$ for all $v \in V(G)$ inductively in topological order and then resolving with the target axioms $\overline{x(z)}_i$, $i \in [d]$. Writing down this resolution proof, one gets the following proposition (which is proven together with Proposition 4.15 below).

Proposition 4.12 ([14]). *For any DAG G with all vertices having indegree 0 or 2, there is a resolution refutation $\pi : Peb^d_G \vdash 0$ in length $L(\pi) = \mathrm{O}\big(d^2 \cdot |V(G)|\big)$ and width $W(\pi) = \mathrm{O}(d)$.*

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

Theorem 4.13 ([11]). *For any DAG G with all vertices having indegree 0 or 2, there is a tree-like resolution refutation π of Peb^1_G such that $L(\pi) = \mathrm{O}(|V(G)|)$ and $Sp(\pi) = \mathrm{O}(1)$.*

Theorem 4.14 ([14]). *For any DAG G with all vertices having indegree 0 or 2, $L_{\mathfrak{T}}(Peb^2_G \vdash 0) = 2^{\Omega(\textsf{Peb}(G))}$.*

As to space, it is not too difficult to see that the black pebbling price of G provides an upper bound for the refutation clause space of Peb^d_G.

Proposition 4.15. *For any DAG G with vertex indegrees 0 or 2, $Sp(Peb^d_G \vdash 0) \leq \textsf{Peb}(G) + \mathrm{O}(1)$.*

[5] This was not known at the time of the original theorems in [31, 47]. What is needed is an explicit construction of superconcentrators of linear density, and it has since been shown how to do this (with [4] apparently being the currently best construction).

Essentially, this is just a matter of combining an optimal black pebbling of G with the resolution refutation idea from [14] sketched above. Since we need the upper bounds on width and space in Propositions 4.12 and 4.15 in the proof of our main theorem, we write down the details for completeness.

Proof of Propositions 4.12 and 4.15. Consider first the bound on space.

Given a black pebbling of G, we construct a resolution 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 $\bigl\{\bigvee_{i=1}^{d} x(v)_i \mid v \in V\bigr\}$ 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 $\mathit{Peb}(G) + \mathrm{O}(1)$.

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 $\overline{x(p)}_i \lor \bigvee_{l=1}^{d} x(r)_l$ for any i can be derived in additional space 3 by resolving $\bigvee_{j=1}^{d} x(q)_j$ with $\overline{x(p)}_i \lor \overline{x(q)}_j \lor \bigvee_{l=1}^{d} x(r)_l$ 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 $\overline{x(p)}_1 \lor \bigvee_{l=1}^{d} x(r)_l$ to get $\bigvee_{i=2}^{d} x(p)_i \lor \bigvee_{l=1}^{d} x(r)_l$, and then resolve this clause with the clauses $\overline{x(p)}_i \lor \bigvee_{l=1}^{d} x(r)_l$ for $i = 2, \ldots, d$ one by one to get $\bigvee_{l=1}^{d} x(r)_l$ in total extra space 4.

It is easy to see that this proof has width $\mathrm{O}(d)$, which proves the claim about width in Proposition 4.12. To get the claim about length, we observe that the subderivation needed when a vertex is black-pebbled has length $\mathrm{O}\bigl(d^2\bigr)$. If we use a pebbling that black-pebbles all vertices once in topological order without ever removing a pebble, we get a refutation in length $L(\pi) = \mathrm{O}\bigl(d^2 \cdot |V(G)|\bigr)$. □

Thus, the refutation clause space of a pebbling contradiction is upper-bounded by the black pebbling price of the underlying DAG. Proposition 4.15 is not quite an optimal strategy with respect to clause space, though. For binary trees [29] improved this bound somewhat to $\mathit{Sp}(\mathit{Peb}^2_{T_h} \vdash 0) \leq \frac{2}{3}h + \mathrm{O}(1)$ by constructing resolution proofs that try to mimic not black pebblings but instead optimal *black-white* pebblings of T_h as presented in [39]. And for one variable per vertex, we know from Theorem 4.13 that $\mathit{Sp}(\mathit{Peb}^1_G \vdash 0) = \mathrm{O}(1)$.

Proving lower bounds on space for pebbling contradictions of degree $d \geq 2$ has turned out to be much harder. For quite some time there was no lower bound on $\mathit{Sp}(\mathit{Peb}^d_G \vdash 0)$ for any DAG G in general resolution (in terms of pebbling price or otherwise). In [29], a lower bound $\mathit{Sp}_{\mathfrak{T}}(\mathit{Peb}^d_{T_h} \vdash 0) = h + \mathrm{O}(1)$ was obtained for the special case of tree-like resolution. Unfortunately, this does not tell us anything about general resolution. For tree-like resolution, if the only way of deriving a clause D is from clauses C_1, C_2 such that $\mathit{Sp}_{\mathfrak{T}}(F \vdash C_i) \geq s$, then it holds that $\mathit{Sp}_{\mathfrak{T}}(F \vdash D) \geq s + 1$ since one of the clauses C_i must be kept in memory while deriving the other clause. This seems to be very different from how general resolution works with respect to space. In [42], the first author showed a lower bound $\mathit{Sp}(\mathit{Peb}^d_{T_h} \vdash 0) = \Omega(h)$ for binary trees and $d \geq 2$, which matches the upper bound up to a constant factor. As the techniques in [42] do not yield anything for more general graphs, this is all that was known prior to this paper.

We now try to present our own intuition for what the correct lower bound on the refutation clause space of pebbling contradictions *should be*. Although the reasoning is quite informal and non-rigorous, our hope is that it will help the reader to navigate the formal proofs that will follow.

As we noted above, the resolution refutation of $Peb^2_{T_h}$ in [29] used to prove the $\frac{2}{3}h + \mathrm{O}(1)$ upper bound for binary tree pebbling contradictions is structurally quite similar to the optimal black-white pebbling of T_h presented in [39], and it somehow feels implausible that any resolution refutation would be able to do significantly better. Also, the lower bound in [42] is proven by relating resolution refutations to black-white pebblings and deriving a lower bound on clause space in terms of pebbling price. This raises the suspicion that the black-white pebbling price $\mathit{BW\text{-}Peb}(G)$ might be a lower bound for $Sp(Peb^d_G \vdash 0)$ also for more general graphs as long as $d \geq 2$.

This suspicion is somewhat strengthened by the fact that for variable space, we do have such a lower bound in terms of black-white pebbling price.[6]

Theorem 4.16 ([11]). *For any $d \in \mathbb{N}^+$, $VarSp(Peb^d_G \vdash 0) \geq \mathit{BW\text{-}Peb}(G)$.*

If the refutation clause space of pebbling contradictions for general DAGs would be constant or very slowly growing, Theorem 4.16 would imply that as $\mathit{BW\text{-}Peb}(G)$ grows larger, the clauses in memory get wider, and thus weaker. Still it would somehow be possible to derive a contradiction from a very small number of these clauses of unbounded width. This appears counterintuitive.

On the other hand, for one variable per vertex, i.e., $d = 1$, refutations of Peb^1_G in constant space have exactly these "counterintuitive" properties. The resolution refutation of Peb^1_G in Theorem 4.13 is constructed by first downloading the pebbling axiom for the target z and then moving the false literals downwards by resolving with pebbling axioms for vertices $v \in V(G) \setminus S$ in reverse topological order. This finally yields a clause $\bigvee_{v \in S} \overline{x(v)}_1 \vee x(z)_1$ of width $|S| + 1$, which can be eliminated by resolving with the source axioms $x(v)_1$ one by one for all $v \in S$ and then with the target axiom $\overline{x(z)}_1$ to yield the empty clause 0.

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

As was argued in Section 2, to prove a lower bound on the refutation clause space of pebbling contradictions it seems natural to try to interpret resolution refutations of Peb^d_G in terms of pebblings of the underlying graph G. Let us say that a vertex v is "true" if $\bigvee_{i=1}^{d} x(v)_i$ has been derived and "false" if $\overline{x(v)}_i$ has been derived for all $i \in [d]$. Any resolution proof refutes a pebbling contradiction by deriving that some vertex v is both true and false and then resolving to get 0. Let w be any vertex with predecessors u, v. Then we can see that if we have derived that u and v are true, by downloading $\overline{x(u)}_i \vee \overline{x(v)}_j \vee \bigvee_{l=1}^{d} x(w)_l$ for all $i, j \in [d]$ we can derive $\bigvee_{l=1}^{d} x(w)_l$. This appears analogous to the rule that if u and v are black-pebbled we can place a black pebble on w. In the opposite direction, if we know $\overline{x(w)}_l$ for all $l \in [d]$, using the axioms $\overline{x(u)}_i \vee \overline{x(v)}_j \vee \bigvee_{l=1}^{d} x(w)_l$ we can derive that either u or v is false. This looks similar to eliminating a white pebble on w by placing white pebbles on the predecessors u and v, and then removing the pebble from w. Generalizing this loose, intuitive reasoning, we argue that a set of black-pebbled vertices V should correspond to the derived conjunction of truth of all $v \in V$, and that a set of white-pebbled vertices W should correspond to the derived disjunction of falsity of some $w \in W$.

Suppose that we could show that as the resolution derivation proceeds, the black and white pebbles corresponding to different clause configurations as outlined above move about on the vertices of G in accordance with the rules of the pebble game. If so, we would get that there is some clause configuration $\mathbb{C}$ corresponding to a lot of pebbles. This could in turn hopefully yield a lower

[6] To be precise, the result in [11] is for $d = 1$, but the proof generalizes easily to any $d \in \mathbb{N}^+$.

bound for the refutation clause space. For if $\mathbb{C}$ corresponds to N black pebbles, i.e., implies N disjoint clauses, it seems likely that $|\mathbb{C}|$ should be linear in N. And if $\mathbb{C}$ corresponds to N white pebbles, $|\mathbb{C}|$ should grow with N if $d \geq 2$, since $\mathbb{C}$ has to force d literals false simultaneously for one out of N vertices.

This is the guiding intuition that served as a starting point for proving the results in this paper. And although quite a few complications arise along the way, we believe that it is important when reading the paper not to let all technical details obscure the rather simple intuitive correspondence sketched above.

5 A Simplified Way of Proving Trade-off Results

Before we launch into the proof of the main result of this paper, however, we quickly present our simplification of the length-space trade-off result in [33], and show how the same ideas can be used to prove other related theorems. We also point out two key ingredients needed for our proofs to work and discuss possible conclusions to be drawn regarding proving trade-off results for resolution. We remark that this section is a somewhat polished write-up of the results previously announced in [43].

We will need the following easy observation.

Observation 5.1. *Suppose that $F = G \wedge H$ where G and H are unsatisfiable CNF formulas over disjoint sets of variables. Then any resolution refutation $\pi : F \vdash 0$ must contain a refutation of either G or H.*

Proof. By induction, we can never resolve a clause derived from G with a clause derived from H, since the sets of variables of the two clauses are disjoint. ☐

5.1 A Proof of Hertel and Pitassi's Trade-off Result

Using the notation in Section 3, and improving the parameters somewhat, the length-variable space trade-off theorem of Hertel and Pitassi [33] can be stated as follows.

Theorem 2.4 (restated). *There is a family of CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

- *The minimal variable space of refuting F_n in resolution is $\mathit{VarSp}(F_n \vdash 0) = \Theta(n)$.*
- *Any resolution refutation $\pi : F_n \vdash 0$ in minimal variable space has length $\exp(\Omega(\sqrt{n}))$.*
- *Adding at most 2 extra units of storage, one can obtain a refutation π' in space $\mathit{VarSp}(\pi') = \mathit{VarSp}(F_n \vdash 0) + 3 = \Theta(n)$ and length $L(\pi') = \mathrm{O}(n)$, i.e., linear in the formula size.*

We note that the CNF formulas used by Hertel and Pitassi, as well as those in our proof, have clauses of width $\Theta(n)$.

Proof of Theorem 2.4. Let G_n be CNF formulas as in Theorem 4.1 having size $\Theta(n)$, refutation length $L(G_n \vdash 0) = \exp(\Omega(n))$ and refutation clause space $\mathit{Sp}(G_n \vdash 0) = \Theta(n)$. Let us define $g(n) = \mathit{VarSp}(G_n \vdash 0)$ to be the refutation variable space of the formulas. Then it holds that $\Omega(n) = g(n) = \mathrm{O}(n^2)$.

Let H_m be the formulas

$$H_m = y_1 \wedge \cdots \wedge y_m \wedge (\overline{y}_1 \vee \cdots \vee \overline{y}_m) \ . \tag{5.1}$$

It is not hard to see that there are resolution refutations $\pi : H_m \vdash 0$ in length $L(\pi) = 2m + 1$ and variable space $\mathit{VarSp}(\pi) = 2m$, and that $L(H_m \vdash 0) = 2m + 1$ and $\mathit{VarSp}(H_m \vdash 0) = 2m$ are also the lower bounds (all clauses must be used in any refutation, and the minimum space refutation must start by downloading the wide clause and some unit clause, and then resolve).

Now define

$$F_n = G_n \wedge H_{\lfloor g(n)/2 \rfloor + 1} \tag{5.2}$$

where G_n and $H_{\lfloor g(n)/2 \rfloor + 1}$ have disjoint sets of variables. By Observation 5.1, any resolution refutation of F_n refutes either G_n or $H_{\lfloor g(n)/2 \rfloor + 1}$. We have

$$\mathit{VarSp}\big(H_{\lfloor g(n)/2 \rfloor + 1} \vdash 0\big) = 2 \cdot (\lfloor g(n)/2 \rfloor + 1) > g(n) = \mathit{VarSp}(G_n \vdash 0) \ , \tag{5.3}$$

so a resolution refutation in minimal variable space must refute G_n in length $\exp(\Omega(n))$. However, allowing at most two more literals in memory, the resolution refutation can disprove the formula $H_{\lfloor g(n)/2 \rfloor + 1}$ instead in length linear in the (total) formula size.

Thus, we have a formula family $\{F_n\}_{n=1}^{\infty}$ of size $\Omega(n) = S(F_n) = \mathrm{O}\big(n^2\big)$ refutable in length and variable space both linear in the formula size, but where any minimum variable space refutation must have length $\exp(\Omega(n))$. Adjusting the indices as needed, we get a formula family with a trade-off of the form stated in Theorem 2.4. ☐

5.2 Some Other Trade-off Results for Resolution

Using a similar trick as in the previous subsection, we can prove the following length-clause space trade-off.

Theorem 1.3 (restated). *There is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

- *The minimal clause space of refuting F_n in resolution is $\mathit{Sp}(F_n \vdash 0) = \Theta\big(\sqrt[3]{n}\big)$.*
- *Any resolution refutation $\pi : F_n \vdash 0$ in minimal clause space must have length $L(\pi) = \exp\big(\Omega\big(\sqrt[3]{n}\big)\big)$.*
- *There are resolution refutations $\pi' : F_n \vdash 0$ in asymptotically minimal clause space $\mathit{Sp}(\pi') = \mathrm{O}\big(\mathit{Sp}(F_n \vdash 0)\big)$ and length $L(\pi') = \mathrm{O}(n)$, i.e., linear in the formula size.*

The same game can be played with refutation width as well.

Theorem 5.2. *There is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

- *The minimal width of refuting F_n is $W(F_n \vdash 0) = \Theta\big(\sqrt[3]{n}\big)$.*
- *Any refutation $\pi : F_n \vdash 0$ in minimal width must have length $L(\pi) = \exp\big(\Omega\big(\sqrt[3]{n}\big)\big)$.*
- *There are refutations $\pi' : F_n \vdash 0$ with $W(\pi') = \mathrm{O}\big(W(F_n \vdash 0)\big)$ and $L(\pi') = \mathrm{O}(n)$.*

We only present the proof of Theorem 1.3, as Theorem 5.2 is proved in exactly the same manner.

Proof of Theorem 1.3. Let G_n be a 3-CNF formula family as in Theorem 4.1 having size $\Theta(n)$, refutation length $L(G_n \vdash 0) = \exp(\Theta(n))$, and refutation clause space $\mathit{Sp}(G_n \vdash 0) = \Theta(n)$. Let H_m be a 3-CNF formula family as in Theorem 4.4 of size $\Theta\big(m^3\big)$ such that $L(H_m \vdash 0) = \mathrm{O}\big(m^3\big)$ and $\mathit{Sp}(H_m \vdash 0) = \Theta(m)$. Define

$$g(n) = \min\big\{m \mid \mathit{Sp}(H_m \vdash 0) > \mathit{Sp}(G_n \vdash 0)\big\} \ . \tag{5.4}$$

Note that since $\mathit{Sp}(H_m \vdash 0) = \Omega(m)$ and $\mathit{Sp}(G_n \vdash 0) = \mathrm{O}(n)$, we know that $g(n) = \mathrm{O}(n)$.

Now as before let $F_n = G_n \wedge H_{g(n)}$, where G_n and $H_{g(n)}$ have disjoint sets of variables. By Observation 5.1, any resolution refutation of F_n is a refutation of either G_n or $H_{g(n)}$. Since $g(n)$ has been chosen so that $\mathit{Sp}\big(H_{g(n)} \vdash 0\big) > \mathit{Sp}(G_n \vdash 0)$, a refutation in minimal clause space has to refute G_n, which requires exponential length. However, since $g(n) = \mathrm{O}(n)$, Theorem 4.4 tells us that there are refutations of $H_{g(n)}$ in length $\mathrm{O}\big(n^3\big)$ and clause space $\mathrm{O}(n)$. ☐

5.3 Making the Main Trick Explicit

The proofs of the theorems in Sections 5.1 and 5.2 come very easily; in fact almost *too* easily. What is it that makes this possible? In this and the next subsection, we want to highlight two key ingredients in the constructions.

The common paradigm for the proofs of Theorems 1.3, 2.4, and 5.2 is as follows. We are given two complexity measures M_1 and M_2 that we want to trade off against one another. We do this by finding formulas G_n and H_m such that

- The formulas G_n are very hard with respect to the first resource measured by M_1, while $M_2(G_n)$ is at most some (more or less trivial) upper bound,
- The formulas H_m are very easy with respect to M_1, but there is some nontrivial *lower* bound on the usage $M_2(H_m)$ of the second resource,
- The index $m = m(n)$ is chosen so as to minimize $M_2(H_{m(n)}) - M_2(G_n) > 0$, i.e., so that $H_{m(n)}$ requires *just a little bit* more of the second resource than G_n.

Then for $F_n = G_n \wedge H_{m(n)}$, if we demand that a resolution refutation π must use the minimal amount of the second resource, it will have to use a large amount of the first resource. However, relaxing the requirement on the second resource by the very small expression $M_2(H_{m(n)}) - M_2(G_n)$, we can get a refutation π' using small amounts of both resources.

Clearly, the formula families $\{F_n\}_{n=1}^{\infty}$ that we get in this way are "redundant" in the sense that each formula F_n is the conjunction of two formulas G_n and H_m which are themselves already unsatisfiable. Formally, we say that a formula F is *minimally unsatisfiable* if F is unsatisfiable, but removing any clause $C \in F$, the remaining subformula $F \setminus \{C\}$ is satisfiable. We note that if we would add the requirement in Sections 5.1 and 5.2 that the formulas under consideration should be minimally unsatisfiable, the proof idea outlined above fails completely. In contrast, the result in [33] seems to be independent of any such conditions. What conclusions can be drawn from this?

On the one hand, trade-off results for minimally unsatisfiable formulas seem more interesting, since they tell us something about a property that some natural formula family has, rather than about some funny phenomena arising because we glue together two totally unrelated formulas.

On the other hand, one could argue that the main motivation for studying space is the connection to memory requirements for proof search algorithms, for instance algorithms using clause learning. And for such algorithms, a minimality condition might appear somewhat arbitrary. There are no guarantees that "real-life" formulas will be minimally unsatisfiable, and most probably there is no efficient way of testing this condition.[7] So in practice, trade-off results for non-minimal formulas might be just as interesting.

5.4 An Auxiliary Trick for Variable Space

A second important reason why our proof of Theorem 2.4 gives sharp results is that we are allowed to use CNF formulas of growing width. It is precisely because of this that we can easily construct the needed formulas H_m that are hard with respect to variable space but easy with respect to length. If we would have to restrict ourselves to k-CNF formulas for k constant, it would be much more difficult to find such examples. Although the formulas in Theorem 4.4 could be plugged in to give a slightly weaker trade-off, we are not aware of any family of k-CNF formulas that can provably give the very sharp result in Theorem 2.4. (Note, though, that the formula families used in the proofs of Theorems 1.3 and 5.2 consist of k-CNF formulas).

[7] The problem of deciding minimal unsatisfiability is NP-hard but not known to be in NP. Formally, a language L is in the complexity class DP if and only if there are two languages $L_1 \in$ NP and $L_2 \in$ co-NP such that $L = L_1 \cap L_2$ [45]. MINIMAL UNSATISFIABILITY is DP-complete [46], and it seems to be commonly believed that DP $\not\subseteq$ NP $\cup$ co-NP.

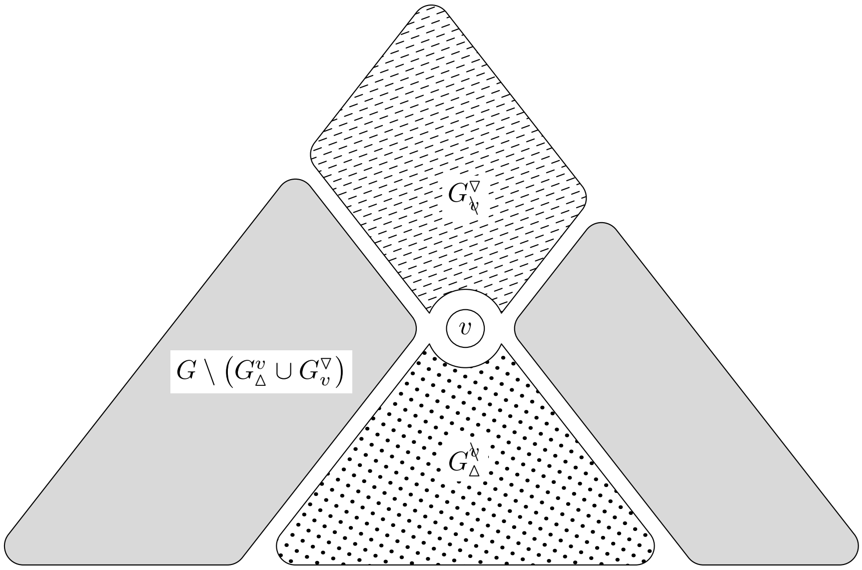


Figure 4: Notation for sets of vertices in DAG G with respect to a vertex v.

This is not the only example of a space measure behaving badly for formulas of growing width. We already discussed the lower bound $Sp(F \vdash 0) \geq W(F \vdash 0) - W(F) + 3$ on clause space in terms of length in Theorem 4.5, and the result in Theorem 4.6 that this inequality is asymptotically strict in the sense that there are k-CNF formula families F_n with $W(F_n \vdash 0) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Theta(\log n)$.

However, if we are allowed to consider formulas of growing width, the fact that the inequality in Theorem 4.5 is not tight is entirely trivial. Namely, let us say that a CNF formula F is *k-wide* if all clauses in F have size at least k. In [28], it was proven that for F a k-wide unsatisfiable CNF formula it holds that $Sp(F \vdash 0) \geq k + 2$. So in order to get a formula family F_n such that $W(F_n \vdash 0) - W(F_n) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \omega(1)$, just pick some suitable formulas $\{F_n\}_{n=1}^{\infty}$ of growing width.

In our opinion, these phenomena are clearly artificial. Since every CNF formula can be rewritten as an equivalent k-CNF formula without increasing the size more than linearly, the right approach when studying space measures in resolution seems to be to require that the formulas under study should have constant width.

As a final comment before moving on to our main result, we note that the open trade-off questions mentioned in Section 11 do not suffer from the technical problems discussed above.

6 A Game for Analyzing Pebbling Contradictions

We now start our construction for 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.

6.1 Some Graph Notation and Definitions

We first present some notation and terminology that will be used in what follows. See Figure 4 for an illustration of the next definition.

Definition 6.1. We let $succ(v)$ denote the immediate successors and $pred(v)$ denote the immediate predecessors of a vertex v in a DAG G. Taking the transivite closures of $succ(\cdot)$ and $pred(\cdot)$, we let $G_v^{\triangledown}$ denote all vertices reachable from v (vertices "above" v) and $G_{\triangle}^{v}$ denote all vertices from which v is reachable (vertices "below" v). We write $G_{\triangle}^{\not v}$ and $G_{\not v}^{\triangledown}$ 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 $u \notin G_{\triangle}^{v}$ and $v \notin G_{\triangle}^{u}$, we say that u and v are *non-comparable* vertices. Otherwise they are *comparable*.

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 $G_{\triangle}^{v}$ and $G_v^{\triangledown}$ are sets of vertices, not subgraphs. However, we will allow ourselves to overload the notation and sometimes use this notation both for the subgraph and its vertices. Moreover, 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.

For our pebble game to work, we require of the graphs under study that they have the following property.

Property 6.2 (Sibling non-reachability). We say that a DAG G has the *Sibling non-reachability property* if for all vertices u and v that are siblings in G, it holds that $u \notin G_{\triangle}^{v}$ and $v \notin G_{\triangle}^{u}$, i.e., the siblings are not reachable from one another.

Phrased differently, Property 6.2 asserts that siblings are non-comparable.

A sufficient condition for Property 6.2 to hold is that if v is reachable from u, then all paths $P : u \rightsquigarrow v$ have the same length. This holds for instance for the class of *layered graphs*, and it is also easy to see directly that layered graphs possess Property 6.2.

Definition 6.3 (Layered DAG). A *layered DAG G* 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 G.

Throughout this paper, we will assume that all source vertices in a layered DAG are located on the bottom level 0. Let us next give a formal definitions of the pyramid graphs that are the focus of this paper.

Definition 6.4 (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 6.5 (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 $u \in G_{\triangle}^{v}$ or $v \in G_{\triangle}^{u}$). 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 let

- $\mathrm{bot}(V)$ denote the bottom vertex of V, i.e., the unique $v \in V$ such that $V \subseteq G_v^{\triangledown}$,
- $\mathrm{top}(V)$ denote the top vertex of V, i.e., the unique $v \in V$ such that $V \subseteq G_{\triangle}^{v}$,
- $\mathfrak{P}_{\mathrm{in}}(V)$ denote the set of all paths $P : \mathrm{bot}(V) \rightsquigarrow \mathrm{top}(V)$ *via V* or *agreeing with V*, i.e., such that $V \subseteq P$, and

- $\mathfrak{P}_{\text{via}}(V)$ denote the set of all source paths *agreeing with* V.

We write $\bigcup \mathfrak{P}_{\text{in}}(V)$ to denote the union of the vertices in all paths $P \in \mathfrak{P}_{\text{in}}(V)$ and $\bigcup \mathfrak{P}_{\text{via}}(V)$ for the union of all vertices in paths $P \in \mathfrak{P}_{\text{via}}(V)$.

In the rest of this paper, we will almost exclusively discuss DAGs with certain structural properties. The next definition is so that we will not have to repeat these properties over and over again.

Definition 6.6 (Blob-pebblable DAG). A *blob-pebblable DAG* is a DAG that has a unique sink, which we will alway denote z, that has vertex indegree 2 for all non-sources, and that satisfies the Sibling non-reachability property 6.2.

6.2 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 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 (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 pebble configurations, each subconfiguration consisting of black pebble together with all the white pebbles this black pebble depends on, and require 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 where 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 ov v, whereas for a blob pebble on V we only know that some vertex $v \in V$ is true, but not which one. For reasons that will perhaps become clearer in Sections 9 and 10, in is natural to consider blobs that are chains (Definition 6.5).

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 v with the help of which we have been able to place the black blob on B. These "associated" or "supporting" white pebbles can be located on any vertex $w \notin B$ that can be visited by a source path P to $\mathrm{top}(B)$ agreeing with B. Formally, the *legal pebble positions* with respect to a chain B with $b = \mathrm{bot}(B)$ is the set of vertices

$$lpp(B) = G^{b}_{\triangle} \cup \Big(\bigcup \mathfrak{P}_{\text{in}}(B) \setminus B\Big) = \bigcup \mathfrak{P}_{\text{via}}(B) \setminus B \ . \tag{6.1}$$

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 6.7 (Blob subconfiguration). For sets of vertices B, W in a blob-pebblable DAG G, $[B]\langle W\rangle$ is a *blob subconfiguration* if $B \neq \emptyset$ is a chain and $W \subseteq lpp(B)$. 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 $\mathbb{S} = \big\{[B_i]\langle W_i\rangle \mid i = 1, \ldots, m\big\}$ together constitute a *blob-pebbling configuration*.

Note in particular that it always holds that $B \cap W = \emptyset$ for a blob subconfiguration $[B]\langle W\rangle$.

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 a vertex v together with white pebbles on its predecessors (unless v is a source, in which case no white pebbles are needed).

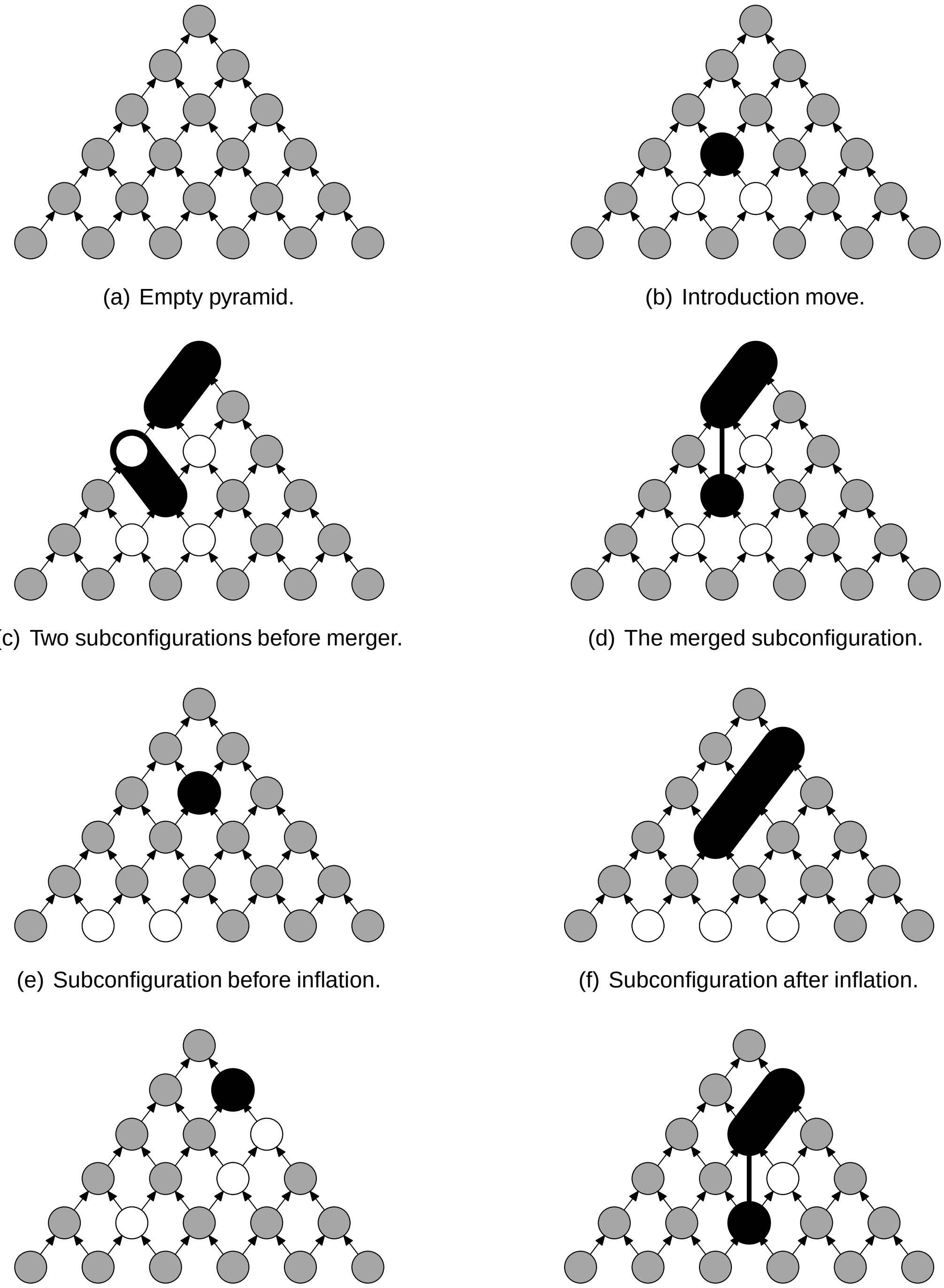
- The analogy for rule 2 for black pebble removal in Definition 3.4 is a rule for "shrinking" black blobs. A vertex v in a blob can be eliminated by *merging* two blob subconfigurations, provided that there is both a black blob and a white pebble on v, and provided that the two black blobs involved in 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, since the white pebble used in the merger is eliminated as well. In addition, however, a white pebble on w can also disappear if its black blob B changes so that w no longer can be visited on a path via B (i.e., if w is no longer a legal pebble position with respect to B).

- 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 5 for some examples of blob-pebbling moves.



(a) Empty pyramid.

(b) Introduction move.

(c) Two subconfigurations before merger.

(d) The merged subconfiguration.

(e) Subconfiguration before inflation.

(f) Subconfiguration after inflation.

(g) Another subconfiguration before inflation.

(h) After inflation with vanished white pebbles.

Figure 5: Examples of moves in the blob-pebble game.

Definition 6.8 (Blob-pebble game). For a blob-pebblable DAG G and 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 configurations such that for all $t \in [\tau]$, $\mathbb{S}_t$ is obtained from $\mathbb{S}_{t-1}$ by one of the following rules:

Introduction $\mathbb{S}_t = \mathbb{S}_{t-1} \cup \{[v]\langle pred(v)\rangle\}$.

Merger $\mathbb{S}_t = \mathbb{S}_{t-1} \cup \{[B]\langle W\rangle\}$ if there are $[B_1]\langle W_1\rangle, [B_2]\langle W_2\rangle \in \mathbb{S}_{t-1}$ such that

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

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 $\mathbb{S}_t = \mathbb{S}_{t-1} \cup \{[B]\langle W\rangle\}$ if there is a $[B']\langle W'\rangle \in \mathbb{S}_{t-1}$ such that

1. $B \supseteq B'$,
2. $B \cap W' = \emptyset$, and
3. $W \supseteq W' \cap lpp(B)$.

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

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

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.

6.3 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 *Blob-Peb*(G):

1. It should be sufficiently high to enable us to prove good lower bounds on *Blob-Peb*(G), preferrably by relating it to the standard black-white pebbling price *BW-Peb*(G).

2. It should also be sufficiently low, so that lower bounds on *Blob-Peb*(G) translate back to lower bounds on the size of the clause configurations.

So when defining pebbling price in Definition 6.9 below, we also have to have in mind the coming Definition 7.2 saying how we will interpret clauses in terms of blobs and pebbles and that these two definitions together should make it possible for us to lower-bound 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 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* among the black blobs, and so as to charge for the subset of supporting white pebbles $W \cap G^b_\Delta$ in a subconfiguration $[B]\langle W \rangle$ that are *located below the bottom vertex* $\mathrm{bot}(B)$ of its black blob B. Multiple distinct blobs with the same bottom vertex come for free, however, and any supporting white pebbles above the bottom vertex of its own blob are also free, although we still have to keep track of them.

Definition 6.9 (Blob-pebbling price). For a subconfiguration $[B]\langle W \rangle$, we say that $\mathcal{B}([B]\langle W \rangle) = \{\mathrm{bot}(B)\}$ is the *chargeable black vertex* and that $\mathcal{W}^\Delta([B]\langle W \rangle) = W \cap G^{\mathrm{bot}(B)}_\Delta$ are the *chargeable white vertices*. The *chargeable vertices* of the subconfiguration $[B]\langle W \rangle$ are all vertices in the union $\mathcal{B}([B]\langle W \rangle) \cup \mathcal{W}^\Delta([B]\langle W \rangle)$. This definition is extended to blob-pebbling configurations $\mathbb{S}$ in the natural way by letting

$$\mathcal{B}(\mathbb{S}) = \bigcup_{[B]\langle W \rangle \in \mathbb{S}} \mathcal{B}([B]\langle W \rangle) = \big\{\mathrm{bot}(B) \mid [B]\langle W \rangle \in \mathbb{S}\big\}$$

and

$$\mathcal{W}^\Delta(\mathbb{S}) = \bigcup_{[B]\langle W \rangle \in \mathbb{S}} \mathcal{W}^\Delta([B]\langle W \rangle) = \bigcup_{[B]\langle W \rangle \in \mathbb{S}} \Big(W \cap G^{\mathrm{bot}(B)}_\Delta\Big) \ .$$

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

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 $\mathbb{S}_\tau = \big\{[B]\langle W \rangle\big\}$. 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.

7 Resolution Derivations Induce Blob-Pebblings

For simplicity, in this section, as well as in the next one, we will write $v_1, \ldots, v_d$ instead of $x(v)_1, \ldots, x(v)_d$ for the d variables associated with v in a dth degree pebbling contradiction. That is, in Sections 7 and 8 small letters with subscripts will denote only variables in propositional logic and nothing else.

It turns out that for technical reasons, it is more natural to ignore the target axioms $\overline{z}_1, \ldots, \overline{z}_d$ 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^d_G. Let us write $^*Peb^d_G = Peb^d_G \setminus \big\{\overline{z}_1, \ldots, \overline{z}_d\big\}$ 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 this pebbling formula $^*Peb_G^d$ instead of refutations of Peb_G^d.

Lemma 7.1. *For any DAG G with sink z, it holds that $Sp(Peb_G^d \vdash 0) = Sp(^*Peb_G^d \vdash \bigvee_{l=1}^{d} z_l)$.*

Proof. For any resolution derivation $\pi^* : {}^*Peb_G^d \vdash \bigvee_{l=1}^{d} z_l$, we can get a resolution refutation of Peb_G^d from π^* in the same space by resolving $\bigvee_{l=1}^{d} z_l$ with all $\overline{z}_l$, $l = 1, \ldots, d$, in space 3.

In the other direction, for $\pi : Peb_G^d \vdash 0$ we can extract a derivation of $\bigvee_{l=1}^{d} z_l$ in at most the same space by simply omitting all downloads of and resolution steps on $\overline{z}_l$ in π, leaving the literals z_l in the clauses. Instead of the final empty clause 0 we get some clause $D \subseteq \bigvee_{l=1}^{d} z_l$, and since $^*Peb_G^d \nvDash D \subsetneqq \bigvee_{l=1}^{d} z_l$ and resolution is sound, we have $D = \bigvee_{l=1}^{d} z_l$. ☐

In view of Lemma 7.1, 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 for instance $V \cup v$ 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, for instance, $[u, v]$ instead of $[\{u, v\}]$.

7.1 Definition of Induced Configurations and Theorem Statement

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

$$Ax^d(r) = \left\{\overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l \mid i, j \in [d]\right\} \tag{7.1}$$

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

$$\mathbb{B}(V) = \left\{\bigvee_{i=1}^{d} v_i \mid v \in V\right\} \tag{7.2}$$

and

$$All^+(V) = \bigvee_{v \in V} \bigvee_{i=1}^{d} v_i \ . \tag{7.3}$$

One can think of $\mathbb{B}(V)$ as "truth of all vertices in V" and $All^+(V)$ 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 $\mathbb{C} \vDash 0$ 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. We remind the reader that the vertex set $lpp(B)$ of legal pebble positions for white pebbles with respect to the chain B was defined in Equation (6.1) on page 26.

Definition 7.2 (Induced blob subconfiguration). Let G be a blob-pebblable DAG and $\mathbb{C}$ a clause configuration derived from $^*Peb_G^d$. Then $\mathbb{C}$ induces the blob subconfiguration $[B]\langle W\rangle$ if there is a clause set $\mathbb{C}_B \subseteq \mathbb{C}$ and a vertex set $S \subseteq G \setminus B$ with $W = S \cap lpp(B)$ such that

$$\mathbb{C}_B \cup \mathbb{B}(S) \vDash All^+(B) \tag{7.4a}$$

but for which it holds for all strict subsets $\mathbb{C}'_B \subsetneqq \mathbb{C}_B$, $S' \subsetneqq S$ and $B' \subsetneqq B$ that

$$\mathbb{C}'_B \cup \mathbb{B}(S) \nvDash All^+(B) \ , \tag{7.4b}$$

$$\mathbb{C}_B \cup \mathbb{B}(S') \nvDash All^+(B) \ , \text{ and} \tag{7.4c}$$

$$\mathbb{C}_B \cup \mathbb{B}(S) \nvDash All^+(B') \ . \tag{7.4d}$$

We write $\mathbb{S}(\mathbb{C})$ to denote the set of all blob subconfigurations induced by $\mathbb{C}$.

To save space, when all conditions (7.4a)–(7.4d) hold, we write

$$\mathbb{C}_B \cup \mathbb{B}(S) \rhd \mathit{All}^+(B) \tag{7.5}$$

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

In the following, we will use the definition of precise implication $\rhd$ also for clauses $\mathit{All}^+(V)$ where the vertex set V is not a chain.

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 $\mathit{All}^+(V) = \bigvee_{v\in V} \bigvee_{i=1}^{d} v_i$ 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.

The main result of this section is as follows.

Theorem 7.3. *Let $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ be a resolution derivation of $\bigvee_{i=1}^{d} z_i$ from $*Peb_G^d$ for a blob-pebblable 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 $\max\{\mathsf{cost}(\mathbb{S}(\mathbb{C}_{t-1})), \mathsf{cost}(\mathbb{S}(\mathbb{C}_t))\} + \mathrm{O}(1)$.*

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

We prove the theorem by forward induction over the derivation π. By the pebbling rules in Definition 6.8, 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 $\mathbb{C}_{t-1} \rightsquigarrow \mathbb{C}_t$. Since the analysis is quite lengthy, we divide it into subsections. But first of all we need some technical lemmas.

7.2 Some Technical Lemmas

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

Lemma 7.4. *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 7.5. *Suppose that C, D are clauses and $\mathbb{C}$ is a set of clauses. Then $\mathbb{C} \cup \big\{C\big\} \vDash D$ if and only if $\mathbb{C} \vDash \overline{a} \lor D$ for all $a \in Lit(C)$.*

Proof. Assume that $\mathbb{C} \cup \big\{C\big\} \vDash D$ and consider any assignment α such that $\alpha(\mathbb{C}) = 1$ and $\alpha(D) = 0$ (if there is no such α, then $\mathbb{C} \vDash D \subseteq \overline{a} \lor D$). Such an α must set C to false, i.e., all $\overline{a}$ to true. Conversely, if $\mathbb{C} \vDash \overline{a} \lor D$ 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 $\alpha(\overline{a} \lor D) = 0$ for some literal $a \in Lit(C)$ satisfied by α. □

Lemma 7.6. *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 $\mathbb{C}_1 = \{C \in \mathbb{C} \mid \exists a \text{ such that } \overline{a} \in Lit(C) \text{ and } a \in Lit(D)\}$ and $\mathbb{C}_2 = \mathbb{C} \setminus \mathbb{C}_1$. Since $\mathbb{C}_2 \nvDash D$ there is an α such that $\alpha(\mathbb{C}_2) = 1$ and $\alpha(D) = 0$. 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 7.7. *Let $\mathbb{C}$ be a clause set derived from $*Peb^d_G$. Suppose that B is a chain and that $S \subseteq G \setminus B$ is a vertex set such that $\mathbb{C} \cup \mathbb{B}(S) \vDash All^+(B)$ and let $W = S \cap lpp(B)$. 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 $\mathbb{C}' \subseteq \mathbb{C}$, $S' \subseteq S$ and $B' \subseteq B$ minimal such that $\mathbb{C}' \cup \mathbb{B}(S') \vDash All^+(B')$. Then $\mathbb{C}' \cup \mathbb{B}(S') \rhd All^+(B')$ by definition. Note, furthermore, that $B' \neq \emptyset$ since the clause set on the left-hand side must be non-contradictory. Also, $\mathbb{C}' \neq \emptyset$ since $B' \cap S' \subseteq B \cap S = \emptyset$, so by Lemma 7.4 it cannot be that $\mathbb{B}(S') \vDash All^+(B')$. This means that $\mathbb{C}$ induces $[B']\langle W'\rangle$ for $W' = S' \cap lpp(B')$. We claim that $[B']\langle W'\rangle$ can be inflated to $[B]\langle W\rangle$, from which the lemma follows.

To verify this claim, note that first two conditions $B' \subseteq B$ and $B \cap W' \subseteq B \cap S = \emptyset$ for inflation moves in Definition 6.8 clearly hold by construction. As to the third condition, we get

$$W' \cap lpp(B) = \big(S' \cap lpp(B')\big) \cap lpp(B) \subseteq S \cap lpp(B) = W$$

which proves the claim. □

We now start the case analysis in the proof of Theorem 7.3 for the different possible derivation steps in a resolution derivation.

7.3 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 $\mathbb{S}(\mathbb{C}_t)$.

7.4 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 move since $\mathbb{C}_{t-1} \subseteq \mathbb{C}_t$. Suppose that $[B]\langle W\rangle$ is a new subconfiguration at time t arising from $\mathbb{C}_B \subseteq \mathbb{C}_{t-1}$ and $S \subseteq G \setminus B$ such that $W = S \cap \mathit{lpp}(B)$ and $\mathbb{C}_B \cup \{C\} \cup \mathbb{B}(S) \rhd \mathit{All}^{+}(B)$. Since C is derived from $\mathbb{C}_{t-1}$, we have $\mathbb{C}_{t-1} \vDash C$. Thus it holds that $\mathbb{C}_{t-1} \cup \mathbb{B}(S) \vDash \mathit{All}^{+}(B)$ and Lemma 7.7 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.

7.5 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 $C \in \mathit{Ax}^d(r)$. Then C must be one of the clauses inducing the subconfiguration, and we get that there are $\mathbb{C}_B \subseteq \mathbb{C}_{t-1}$ and $S \subseteq G \setminus B$ with $W = S \cap \mathit{lpp}(B)$ such that

$$\mathbb{C}_B \cup \{C\} \cup \mathbb{B}(S) \rhd \mathit{All}^{+}(B) \ . \tag{7.6}$$

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

We do the proof by a case analysis over r depending on where in the graph this vertex is located in relation to B. To simplify the proofs for the different cases, we first show a general technical lemma about pebble induction at axiom download.

Lemma 7.8. *Suppose that $\mathbb{C}_t = \mathbb{C}_{t-1} \cup C$ for an axiom $C \in \mathit{Ax}^d(r)$ and that $[B]\langle W\rangle$ is a new blob subconfiguration induced at time t as witnessed by* (7.6). *Then it holds that:*

1. $r \notin S$.
2. $\mathit{pred}(r) \cap B = \emptyset$.
3. *If $r \notin B$, then $\mathbb{C}_{t-1}$ induces $[B]\langle W \cup (\{r\} \cap \mathit{lpp}(B))\rangle$ if r is a source, and otherwise this subconfiguration can be derived from $\mathbb{S}(\mathbb{C}_{t-1})$ by inflation.*
4. *If r is a non-source vertex and $v \in \mathit{pred}(r)$ is such that $v \in \mathit{lpp}(B) \setminus S$, then we can derive $[B \cup v]\langle S \cap \mathit{lpp}(B \cup v)\rangle$ from $\mathbb{S}(\mathbb{C}_{t-1})$ by inflation.*

Proof. Suppose that $[B]\langle W\rangle \in \mathbb{S}(\mathbb{C}_t) \setminus \mathbb{S}(\mathbb{C}_{t-1})$. For part 1, noting that $\mathbb{B}(r) \vDash C$ for $C \in \mathit{Ax}^d(r)$ we see that $r \notin S$, as otherwise the implication (7.6) cannot be precise since C can be omitted.

If r is a source part 2 is trivial, so suppose $\mathit{pred}(r) = \{p, q\}$ and $C = \overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l$. Then it follows from Lemma 7.6 that $\{p, q\} \cap B = \emptyset$.

For part 3, if r is a source, we have $C = \bigvee_{i=1}^{d} r_i$ and (7.6) becomes

$$\mathbb{C}_B \cup \mathbb{B}(S \cup r) \rhd \mathit{All}^{+}(B) \tag{7.7}$$

for $S \cup r \subseteq G \setminus B$, which shows that $\mathbb{C}_{t-1}$ induces

$$\begin{aligned} [B]\langle (S \cup r) \cap \mathit{lpp}(B)\rangle &= [B]\langle (S \cap \mathit{lpp}(B)) \cup (r \cap \mathit{lpp}(B))\rangle \\ &= [B]\langle (W \cup (r \cap \mathit{lpp}(B))\rangle \ . \end{aligned} \tag{7.8}$$

If r is a non-source we do not get a precise implication but still have

$$\mathbb{C}_B \cup \mathbb{B}(S \cup r) \vDash All^+(B) \tag{7.9}$$

and Lemma 7.7 yields that $[B]\langle (S \cup r) \cap \mathit{lpp}(B)\rangle = [B]\langle W \cup (r \cap \mathit{lpp}(B))\rangle$ is derivable by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$.

If $v \in \mathit{pred}(r)$ in part 4, the downloaded axiom can be written on the form $C = C' \vee \overline{v}_i$. Applying Lemma 7.5 on (7.6) we get

$$\mathbb{C}_B \cup \mathbb{B}(S) \vDash All^+(B) \vee v_i \subseteq All^+(B \cup v) \ . \tag{7.10}$$

By assumption, we have that $B \cup v$ is a chain and that $S \subseteq G \setminus (B \cup v)$, so Lemma 7.7 says that $[B \cup v]\langle S \cap \mathit{lpp}(B \cup v)\rangle$ is derivable from $\mathbb{S}(\mathbb{C}_{t-1})$ by inflation. □

What we get from Lemma 7.8 is not in itself sufficient to derive the new blob subconfiguration $[B]\langle W\rangle$ in the blob-pebble game, but the lemma provides subconfigurations that will be used as building blocks in the derivations of $[B]\langle W\rangle$ below.

Now we are ready for the case analysis over the vertex r for the downloaded axiom clause $C \in Ax^d(r)$. Recall that the assumption is that there exists a blob subconfiguration $[B]\langle W\rangle \in \mathbb{S}(\mathbb{C}_t) \setminus \mathbb{S}(\mathbb{C}_{t-1})$ induced through (7.6) for $\mathbb{C}_B \subseteq \mathbb{C}_{t-1}$ and $S \subseteq G \setminus B$ with $W = S \cap \mathit{lpp}(B)$. Remember also that we want to explain all new subconfigurations in $\mathbb{S}(\mathbb{C}_t) \setminus \mathbb{S}(\mathbb{C}_{t-1})$ in terms of pebbling moves from $\mathbb{S}(\mathbb{C}_t) \cup \{[r]\langle \mathit{pred}(r)\rangle\}$. As illustrated in Figure 6, the cases for r are:

1. $r \in G \setminus \left(G^b_\Delta \cup \bigcup \mathfrak{P}_{\text{in}}(B)\right)$ for $b = \mathrm{bot}(B)$,
2. $r \in \bigcup \mathfrak{P}_{\text{in}}(B) \setminus B$,
3. $r \in B \setminus \{b\}$ for $b = \mathrm{bot}(B)$,
4. $r = \mathrm{bot}(B)$, and
5. $r \in G^b_\Delta$ for $b = \mathrm{bot}(B)$.

7.5.1 Case 1: $r \in G \setminus \left(G^b_\Delta \cup \bigcup \mathfrak{P}_{\text{in}}(B)\right)$ for $b = \mathrm{bot}(B)$

If $r \in G \setminus \left(G^b_\Delta \cup \bigcup \mathfrak{P}_{\text{in}}(B)\right)$, this means that the vertex r is outside the set of vertices covered by source paths via B to $\mathrm{top}(B)$. In other words, $r \notin \mathit{lpp}(B) \cup B$ and part 3 of Lemma 7.8 yields that $\big[B\big]\big\langle W \cup (r \cap \mathit{lpp}(B))\big\rangle = [B]\langle W\rangle$ is derivable from $\mathbb{S}(\mathbb{C}_{t-1})$ by inflation. Note that we need no intermediate subconfigurations in this case.

7.5.2 Case 2: $r \in \bigcup \mathfrak{P}_{\text{in}}(B) \setminus B$

This is the first more challenging case, and we do it in some detail to show how the reasoning goes. The proofs for the rest of the cases are analogous and will be presented in slightly more condensed form.

The condition $r \in \bigcup \mathfrak{P}_{\text{in}}(B) \setminus B$ says that the vertex r is located on some path from $\mathrm{bot}(B)$ via B to $\mathrm{top}(B)$ strictly above the bottom vertex $b = \mathrm{bot}(B)$. In particular, this means that r cannot be a source vertex. Let $\mathit{pred}(r) = \{p, q\}$ and denote the downloaded axiom clause $C = \overline{p}_i \vee \overline{q}_j \vee \bigvee_{l=1}^{d} r_l$.

Part 3 of Lemma 7.8 says that we can derive the blob subconfiguration

$$[B]\langle W \cup (r \cap \mathit{lpp}(B))\rangle = [B]\langle W \cup r\rangle \tag{7.11}$$

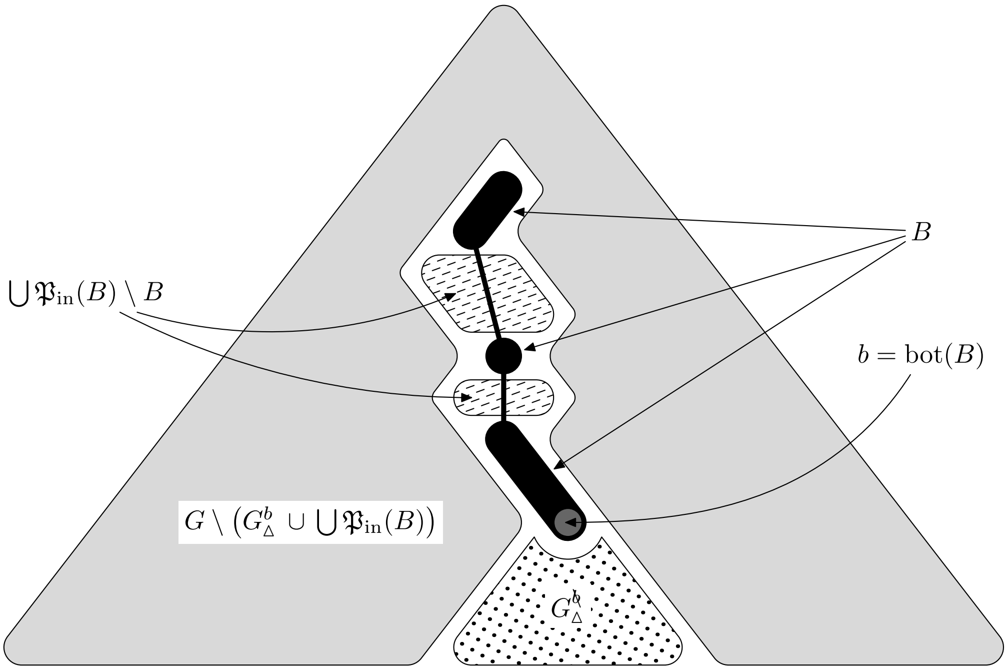


Figure 6: Cases for vertex r with respect to new black blob B at download of axiom $C \in Ax^{d}(r)$.

by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$, where the equality holds since $r \in \bigcup \mathfrak{P}_{\text{in}}(B) \setminus B \subseteq lpp(B)$ by Definition 6.7. Also, since r is on some path above b, at least one of the predecessors of r must be located on some path from b as well. That is, translating what was just said into our notation we have that the fact that $r \in \bigcup \mathfrak{P}_{\text{in}}(B) \cap G^{\triangledown}_{\not{b}}$ implies that either $p \in \bigcup \mathfrak{P}_{\text{in}}(B)$ or $q \in \bigcup \mathfrak{P}_{\text{in}}(B)$ or both. By symmetry, we get two cases: $p \in \bigcup \mathfrak{P}_{\text{in}}(B),\ q \notin \bigcup \mathfrak{P}_{\text{in}}(B)$ and $\{p, q\} \subseteq \bigcup \mathfrak{P}_{\text{in}}(B)$. Let us look at them in order.

I. $p \in \bigcup \mathfrak{P}_{\text{in}}(B),\ q \notin \bigcup \mathfrak{P}_{\text{in}}(B)$: We make a subcase analysis depending on whether $p \in B \cup W$ or not. Recall from part 2 of Lemma 7.8 that $p \notin B$. The two remaining cases are $p \in W$ and $p \notin B \cup W$.

   (a) $p \in W$: Let v be the uppermost vertex in B below p, or in formal notation

$$v = \text{top}(G^{p}_{\Delta} \cap B) \ . \tag{7.12}$$

   Such a vertex v must exist since $p \in \bigcup \mathfrak{P}_{\text{in}}(B) \setminus B$. Since p is above v and is a predecessor of r, it lies on some path from v to r, i.e., $p \in \bigcup \mathfrak{P}_{\text{in}}(\{v, r\}) \setminus \{v, r\}$. For the sibling q we have $q \notin \bigcup \mathfrak{P}_{\text{in}}(\{v, r\})$. This is so since $q \notin \bigcup \mathfrak{P}_{\text{in}}(B)$ and for any path $P \in \mathfrak{P}_{\text{in}}(\{v, r\})$ it holds that $P \subseteq \bigcup \mathfrak{P}_{\text{in}}(B)$ since there is nothing inbetween v and r in B, i.e., $\left(\bigcup \mathfrak{P}_{\text{in}}(\{v, r\}) \setminus \{v, r\}\right) \cap B = \emptyset$. Also, $q \notin G^{\not{p}}_{\Delta} \supseteq G^{\not{v}}_{\Delta}$ because of the Sibling non-reachability property 6.2. Hence, it must hold that $q \notin lpp(\{v, r\})$.

   We can use this information to make blob-pebbling moves resulting in $[B]\langle W \rangle$ as follows. First introduce $[r]\langle p, q \rangle$ and inflate this subconfiguration to

$$[v, r]\langle \{p, q\} \cap lpp(\{v, r\}) \rangle = [v, r]\langle p \rangle \ . \tag{7.13}$$

   Then derive the subconfiguration $[B]\langle W \cup r \rangle$ in (7.11) by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$. Finally, merge the two subconfigurations (7.11) and (7.13). The result of this merger move is $[B \cup v]\langle W \cup p \rangle = [B]\langle W \rangle$.

(b) $p \notin B \cup W$: Note that $p \in \mathfrak{P}_{\text{in}}(B) \setminus B$ by assumption. Also, it must hold that $p \notin S$ since otherwise we would get the contradiction $p \in S \cap (\mathfrak{P}_{\text{in}}(B) \setminus B) \subseteq S \cap \mathit{lpp}(B) = W$. Thus, $p \in \mathit{lpp}(B) \setminus S$ and part 4 of Lemma 7.8 yields that we can derive the blob subconfiguration

$$[B \cup p]\langle W_p \rangle \text{ for } W_p \subseteq W \tag{7.14}$$

by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$, where $W_p = S \cap \mathit{lpp}(B \cup p) \subseteq S \cap \mathit{lpp}(B) = W$ since $\mathit{lpp}(B \cup p) \subseteq \mathit{lpp}(B)$ if $p \in \bigcup \mathfrak{P}_{\text{in}}(B)$. (This last claim is easily verified directly from Definition 6.7.)

With $v = \text{top}(G^p_\Delta \cap B)$ as in (7.12), introduce $[r]\langle p, q \rangle$ and inflate to $[v, r]\langle p \rangle$ as in (7.13). Merging the subconfigurations (7.13) and (7.14) yields

$$[B \cup \{v, r\}]\langle W_p \rangle = [B \cup r]\langle W_p \rangle \tag{7.15}$$

and a second merger of the resulting subconfiguration (7.15) with the subconfiguration in (7.11) produces $[B]\langle W \cup W_p \rangle = [B]\langle W \rangle$.

This finishes the case $p \in \bigcup \mathfrak{P}_{\text{in}}(B)$, $q \notin \bigcup \mathfrak{P}_{\text{in}}(B)$.

II. $\{p, q\} \subseteq \bigcup \mathfrak{P}_{\text{in}}(B)$: By part 2 of Lemma 7.8 $\{p, q\} \cap B = \emptyset$, so $\{p, q\} \subseteq \mathfrak{P}_{\text{in}}(B) \setminus B$. By symmetry, we have the following subcases for p and q with respect to membership in B and W.

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

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

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

We analyze these subcases one by one.

(a) $\{p, q\} \subseteq W$: This is easy. Just introduce $[r]\langle p, q \rangle$ and merge this subconfiguration with the subconfiguration (7.11) to get $[B]\langle W \cup \{p, q\} \rangle = [B]\langle W \rangle$.

(b) $p \in W,\ q \notin W$: In this case it must hold that $q \notin S$ since otherwise we would have $q \in S \cap (\mathfrak{P}_{\text{in}}(B) \setminus B) \subseteq S \cap \mathit{lpp}(B) = W$ contradicting the assumption. Thus $q \in (\mathfrak{P}_{\text{in}}(B) \setminus B) \setminus S \subseteq \mathit{lpp}(B) \setminus S$ and part 4 of Lemma 7.8 allows us to derive

$$[B \cup q]\langle W_q \rangle \text{ for } W_q \subseteq W \tag{7.16}$$

by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$. Here we have $W_q = S \cap \mathit{lpp}(B \cup q) \subseteq S \cap \mathit{lpp}(B) = W$ since $\mathit{lpp}(B \cup q) \subseteq \mathit{lpp}(B)$ when $q \in \bigcup \mathfrak{P}_{\text{in}}(B)$.

Introduce $[r]\langle p, q \rangle$ and merge with the subconfiguration (7.16) to get

$$[B \cup r]\langle W_q \cup p \rangle \tag{7.17}$$

and then merge (7.17) with $[B]\langle W \cup r \rangle$ from (7.11) to get $[B]\langle W \cup W_q \cup p \rangle = [B]\langle W \rangle$.

(c) $\{p, q\} \cap B \cup W = \emptyset$: Just as for the vertex q in case case IIb, here it holds for both p and q that $\{p, q\} \subseteq \mathit{lpp}(B) \setminus S$. Part 4 of Lemma 7.8 yields subconfigurations $[B \cup p]\langle W_p \rangle$ for $W_p \subseteq W$ as in (7.14) and $[B \cup q]\langle W_q \rangle$ for $W_q \subseteq W$ as in (7.16) derived by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$.

Introduce $[r]\langle p, q \rangle$ and merge with (7.14) on p to get

$$[B \cup r]\langle W_p \cup q \rangle \tag{7.18}$$

and then merge (7.18) with (7.16) on q resulting in

$$[B \cup r]\langle W_p \cup W_q \rangle \ . \tag{7.19}$$

Finally, merge (7.19) with (7.11) on r to get $[B]\langle W \cup W_p \cup W_q \rangle = [B]\langle W \rangle$.

This concludes the case $r \in \bigcup \mathfrak{P}_{\text{in}}(B) \setminus B$. We can see that in all subcases, the new blob subconfiguration $[B]\langle W \rangle$ is derivable from $\mathbb{S}(\mathbb{C}_{t-1}) \cup [r]\langle \mathit{pred}(r) \rangle$ by inflation moves followed by mergers on some subset of $\{p, q, r\}$.

Let us analyze the cost of deriving $[B]\langle W \rangle$. We want to bound the cost of the intermediate subconfigurations that are used in the transition from $\mathbb{S}(\mathbb{C}_{t-1})$ to $\mathbb{S}(\mathbb{C}_t)$ but are not present in $\mathbb{S}(\mathbb{C}_t)$. We first note that for the subconfigurations $[B]\langle W \cup r \rangle$, $[B \cup p]\langle W_p \rangle$, $[B \cup q]\langle W_q \rangle$ and $[B \cup r]\langle W' \rangle$ for various $W' \subseteq W$, the chargeable vertices are all subsets of the chargeable vertices of the final subconfiguration $[B]\langle W \rangle$. This is so since $b = \mathrm{bot}(B)$ is the bottom vertex in all these black blobs, and all chargeable white vertices are contained in $W \cap G^b_\vartriangle$. The subconfigurations $[r]\langle p, q \rangle$ and $[v, r]\langle p \rangle$ for $v = \mathrm{top}(G^p_\vartriangle \cap B)$ can incur an extra cost, however, but this cost is clearly bounded by $|\{p, q, r, v\}| = 4$.

7.5.3 Case 3: $r \in B \setminus \{b\}$ for $b = \mathrm{bot}(B)$

First we note that in this case, we can no longer use part 3 of Lemma 7.8 to derive the blob subconfiguration $[B]\langle W \cup r \rangle$ of (7.11). The vertex r cannot be added to the support S since it is contained in B. Also, we note that r cannot be a source since it is above the bottom vertex b. As usual, let us write $\mathit{pred}(r) = \{p, q\}$.

Observe that just as in case 2 (Section 7.5.2) we must have either $p \in \bigcup \mathfrak{P}_{\text{in}}(B)$ or $q \in \bigcup \mathfrak{P}_{\text{in}}(B)$ or both. By symmetry we get the same two cases for membership of p and q in $\bigcup \mathfrak{P}_{\text{in}}(B)$, namely $p \in \bigcup \mathfrak{P}_{\text{in}}(B),\ q \notin \bigcup \mathfrak{P}_{\text{in}}(B)$ and $\{p, q\} \subseteq \bigcup \mathfrak{P}_{\text{in}}(B)$.

I. $p \in \bigcup \mathfrak{P}_{\text{in}}(B),\ q \notin \bigcup \mathfrak{P}_{\text{in}}(B)$: As before, $p \notin B$ by part 2 of Lemma 7.8. We make a subcase analysis depending on whether $p \in W$ or $p \notin B \cup W$.

   As in (7.12) we let $v = \mathrm{top}(G^p_\vartriangle \cap B)$ and note that $p \in \bigcup \mathfrak{P}_{\text{in}}(\{v, r\}) \setminus \{v, r\}$. For q we have $q \notin \bigcup \mathfrak{P}_{\text{in}}(\{v, r\})$ since $q \notin \bigcup \mathfrak{P}_{\text{in}}(B)$ but $\{v, r\} \subseteq \bigcup \mathfrak{P}_{\text{in}}(B)$ and there is nothing inbetween v and r in B. Also, $q \notin G^{\not p}_\vartriangle \supseteq G^{\not v}_\vartriangle$ because of the Sibling non-reachability property 6.2. Hence, it holds that $q \notin \mathit{lpp}(\{v, r\})$.

   (a) $p \in W$: Introduce $[r]\langle p, q \rangle$, inflate $[r]\langle p, q \rangle$ to $[v, r]\langle \{p, q\} \cap \mathit{lpp}(\{v, r\}) \rangle = [v, r]\langle p \rangle$ as in (7.13) and continue the inflation to $[B \cup \{v, r\}]\langle W \cup p \rangle = [B]\langle W \rangle$.

   (b) $p \notin B \cup W$: Just as in case 2, $p \notin W$ implies $p \notin S$, so $p \in \mathit{lpp}(B) \setminus S$ and we can use part 4 of Lemma 7.8 to derive $[B \cup p]\langle W_p \rangle$ for $W_p \subseteq W$ as in (7.14). Introduce $[r]\langle p, q \rangle$, inflate to $[v, r]\langle p \rangle$ as in (7.13) and merge (7.13) and (7.14) on p resulting in $[B \cup \{v, r\}]\langle W_p \rangle = [B]\langle W_p \rangle$, which can be inflated to $[B]\langle W \rangle$.

II. $\{p, q\} \subseteq \bigcup \mathfrak{P}_{\text{in}}(B)$: We have the same possibilities to consider for containment of p and q in $B \cup W$ as in case 2(II) on page 36.

   (a) $\{p, q\} \subseteq W$: This is immediate. Introduce the subconfiguration $[r]\langle p, q \rangle$ and inflate to $[B \cup r]\langle W \cup \{p, q\} \rangle = [B]\langle W \rangle$.

   (b) $p \in W,\ q \notin B \cup W$: Apply part 4 of Lemma 7.8 to derive $[B \cup q]\langle W_q \rangle$ for $W_q \subseteq W$ by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$. Then introduce $[r]\langle p, q \rangle$ and merge on q to get the subconfiguration $[B \cup r]\langle W_q \cup p \rangle = [B]\langle W_q \cup p \rangle$, which can be inflated further to $[B]\langle W_q \cup p \cup W \rangle = [B]\langle W \rangle$.

(c) $\{p, q\} \cap (B \cup W) = \emptyset$: In the same way as in case IIb, derive the subconfigurations $[B \cup p]\langle W_p \rangle$ and $[B \cup q]\langle W_q \rangle$ with $W_p \cup W_q \subseteq W$ from $\mathbb{S}(\mathbb{C}_{t-1})$ by inflation. Introduce $[r]\langle p, q \rangle$ and merge twice, first on p and then on q, to get $[B]\langle W_p \cup W_q \rangle$, which can be inflated to $[B]\langle W \rangle$.

This concludes the case $r \in B \setminus \{b\}$. We see that in all subcases the new blob subconfiguration $[B]\langle W \rangle$ is derivable from $\mathbb{S}(\mathbb{C}_{t-1}) \cup [r]\langle \mathit{pred}(r) \rangle$ by inflation moves followed by mergers on some subset of $\{p, q\}$, possibly followed by one more inflation move.

As in the previous case, the bottom vertex in all of the black blobs $[B \cup p]$, $[B \cup q]$ and $[B \cup r]$ is $b = \mathrm{bot}(B)$, and the corresponding chargeable white pebbles are subsets of those of W. The extra cost caused by the subconfigurations $[r]\langle p, q \rangle$ and $[v, r]\langle p \rangle$ is at most 4.

7.5.4 Case 4: $r = \mathrm{bot}(B)$

If r is a source, any $[B]\langle W \rangle$ with $r \in B$ can be derived by introducing $[r]\langle \mathit{pred}(r) \rangle = [r]\langle \emptyset \rangle$ and inflating. Suppose therefore that $r = \mathrm{bot}(B)$ is not a source and let $\mathit{pred}(r) = \{p, q\}$. Then it holds that $\{p, q\} \subseteq G^{r}_{\triangle} \subseteq \mathit{lpp}(B)$, i.e., the vertex sets $B \cup p$ and $B \cup q$ are both chains.

By symmetry, we have three cases for p and q with respect to membership in W. (It is still true that $\{p, q\} \cap B = \emptyset$ by part 2 of Lemma 7.8.)

(a) $\{p, q\} \subseteq W$: Immediate. Introduce $[r]\langle p, q \rangle$ and inflate to $[B \cup r]\langle W \cup \{p, q\} \rangle = [B]\langle W \rangle$.

(b) $p \in W,\ q \notin W$: Enlist the help of our old friend Lemma 7.8, part 4, to derive $[B \cup q]\langle W_q \rangle$ for $W_q \subseteq W$ by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$ (where $W_q \subseteq W$ holds since $\mathit{lpp}(B \cup v) \subseteq \mathit{lpp}(B)$ if $v \in G^{b}_{\triangle}$). Introduce $[r]\langle p, q \rangle$ and merge with $[B \cup q]\langle W_q \rangle$ to get $[B \cup r]\langle W_q \cup p \rangle = [B]\langle W_q \cup p \rangle$. Then inflate $[B]\langle W_q \cup p \rangle$ to $[B]\langle W_q \cup p \cup W \rangle = [B]\langle W \rangle$.

(c) $\{p, q\} \cap W = \emptyset$: Following an established tradition, mimic case b and derive $[B \cup p]\langle W_p \rangle$ and $[B \cup q]\langle W_q \rangle$ with $W_p \cup W_q \subseteq W$ by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$. Introduce $[r]\langle p, q \rangle$, do two mergers to get $[B]\langle W_p \cup W_q \rangle$ and inflate to $[B]\langle W \rangle$.

This takes care of the case $r = b$. Again, in all subcases our new subconfiguration $[B]\langle W \rangle$ is derivable from $\mathbb{S}(\mathbb{C}_{t-1}) \cup [r]\langle \mathit{pred}(r) \rangle$ by inflation moves followed by mergers on some subset of $\{p, q\}$, possibly followed by one more inflation move.

This time the blobs $[B \cup p]$ and $[B \cup q]$ can cause an extra intermediate cost of 1 each for the bottom vertices p and q, and $[r]\langle p, q \rangle$ potentially adds an extra cost 1 for r, giving that the intermediate extra cost is bounded by 3.

7.5.5 Case 5: $r \in G^{b}_{\triangle}$ for $b = \mathrm{bot}(B)$

This final case is very similar to the previous case $r = \mathrm{bot}(B)$. Note first that $r \in G^{b}_{\triangle} \subseteq \mathit{lpp}(B)$. If r is a source, then $C = \bigvee_{i=1}^{d} r_i$ and we have

$$\mathbb{C}_B \cup \{C\} \cup \mathbb{B}(S) = \mathbb{C}_B \cup \mathbb{B}(S \cup r) \rhd \mathit{All}^{+}(B) \tag{7.20}$$

at time $t - 1$, which shows that $[B]\langle W \cup r \rangle \in \mathbb{S}(\mathbb{C}_{t-1})$. Hence, we can introduce $[r]\langle \mathit{pred}(r) \rangle = [r]\langle \emptyset \rangle$ and merge on r to get $[B]\langle W \rangle$.

As usual, the more interesting case is when r is a non-source with $\mathit{pred}(r) = \{p, q\}$. The case analysis is just as in case 4 (Section 7.5.4). However, note that now we can again use part 3 of Lemma 7.8 to derive $[B]\langle W \cup r \rangle$ from $\mathbb{S}(\mathbb{C}_{t-1})$ by inflation since it holds that $r \notin B$.

(a) $\{p, q\} \subseteq W$: Introducing $[r]\langle p, q \rangle$ and merging with $[B]\langle W \cup r \rangle$ yields $[B]\langle W \rangle$.

(b) $p \in W,\ q \notin W$: Appeal to part 4 of Lemma 7.8 to get $[B \cup q]\langle W_q \rangle$ for $W_q \subseteq W$ by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$. Introduce $[r]\langle p, q\rangle$ and merge to get $[B \cup r]\langle W_q \cup p\rangle$, and merge again with $[B]\langle W \cup r\rangle$ to get $[B]\langle W\rangle$.

(c) $\{p, q\} \cap W = \emptyset$: As in case b above for q, derive $[B \cup p]\langle W_p\rangle$ and $[B \cup q]\langle W_q\rangle$ with $W_p \cup W_q \subseteq W$ by inflation from $\mathbb{S}(\mathbb{C}_{t-1})$. Introduce $[r]\langle p, q\rangle$ and do two mergers to get $[B \cup r]\langle W_p \cup W_q\rangle$. Finally merge $[B \cup r]\langle W_p \cup W_q\rangle$ with $[B]\langle W \cup r\rangle$ to get $[B]\langle W\rangle$.

This takes care of the case $r = G^{b}_{\Delta}$. We note that in all subcases of this case, $[B]\langle W\rangle$ is derivable from $\mathbb{S}(\mathbb{C}_{t-1}) \cup [r]\langle pred(r)\rangle$ by inflation moves followed by mergers on some subset of $\{p, q, r\}$. Again, the extra intermediate pebbling cost is bounded by $|\{p, q, r\}| = 3$.

7.6 Wrapping up the Proof

If $\pi = \{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is a derivation of $\bigvee_{i=1}^{d} z_i$ from $*Peb^d_G$, it is easily verified from Definition 7.2 that $\mathbb{S}(\mathbb{C}_0) = \mathbb{S}(\emptyset) = \emptyset$ and $\mathbb{S}(\mathbb{C}_\tau) = \mathbb{S}(\{\bigvee_{i=1}^{d} z_i\}) = \{[z]\langle\emptyset\rangle\}$.

In Sections 7.3, 7.4, and 7.5, we have shown how to do the intermediate blob-pebbling moves to get from $\mathbb{S}(\mathbb{C}_{t-1})$ to $\mathbb{S}(\mathbb{C}_t)$ in the case of erasure, inference and axiom download, respectively. For erasure and inference, the blob-pebbling cost changes monotonically during the transition $\mathbb{S}(\mathbb{C}_{t-1}) \rightsquigarrow \mathbb{S}(\mathbb{C}_t)$. In the case of axiom download, there can be an extra cost of 4 incurred for deriving each $[B]\langle W\rangle \in \mathbb{S}(\mathbb{C}_t) \setminus \mathbb{S}(\mathbb{C}_{t-1})$. We have no a priori upper bound on $\left|\mathbb{S}(\mathbb{C}_t) \setminus \mathbb{S}(\mathbb{C}_{t-1})\right|$, but if we just derive the new subconfigurations one by one and erase all intermediate subconfigurations inbetween these derivations, we will keep the total extra cost below 4.

This shows that the complete blob-pebbling $\mathcal{P}_\pi$ of G associated to a resolution derivation $\pi : *Peb^d_G \vdash \bigvee_{i=1}^{d} z_i$ by the construction in this section has blob-pebbling cost bounded from above by $\mathit{cost}(\mathcal{P}_\pi) \leq \max_{\mathbb{C}\in\pi}\{\mathit{cost}(\mathbb{S}(\mathbb{C}))\} + 4$. Theorem 7.3 is thereby proven.

8 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 7.2, then the cost of $\mathbb{S}(\mathbb{C})$ as specified in Definition 6.9 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 8.5 below.

Note that we cannot expect a proof of this fact to work regardless of the pebbling degree d. The induced blob-pebbling in Section 7 makes no assumptions about d, but for first-degree pebbling contradictions we know that $Sp(*Peb^1_G \vdash z_1) = Sp(Peb^1_G \vdash 0) = \mathrm{O}(1)$. Provided $d \geq 2$, though, we show that one has to pay at least $|\mathbb{C}| \geq N$ clauses to get an induced blob-pebbling configuration of cost N.

We introduce some notation to simply 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

$$V(C) = \{u \in V(G) \,|\, Vars^d(u) \cap Vars(C) \neq \emptyset\} \tag{8.1}$$

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

$$\mathbb{C}[\![U]\!] = \{C \in \mathbb{C} \mid V(C) \cap U \neq \emptyset\} \tag{8.2}$$

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 8.5. Intuitively, we will use Lemma 8.1 below together with Lemma 7.4 on page 32 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 8.1. *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, for instance, [1, 6, 20, 38]). 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 8.2. *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 7.4.

The inductive step just generalizes the proof of Lemma 7.4. 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 $|F_V| > |V|$. The theorem follows by induction. □

Continuing our intuitive argument, given that Lemmas 7.4 and 8.1 tell us that many induced subconfigurations implies the presence of many variables in $\mathbb{C}$, we will use Theorem 8.2 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 $\mathbb{B}(V) = \left\{ \bigvee_{i \in [d]} v_i \middle| v \in V \right\}$ and $All^+(V) = \bigvee_{v \in V} \bigvee_{i \in [d]} v_i$ from Section 7.

Example 8.3. 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

$$\mathbb{C} \vDash All^+(B_i) \tag{8.3}$$

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 $\mathbb{C}$ satisfying (8.3) for $i = 1, \ldots, N$ is quite simply the set

$$\mathbb{C} = \left\{ All^+(B_i) \middle| i = 1, \ldots N \right\} \tag{8.4}$$

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 7.4 says that there will be extra clauses needed to "neutralize" the literals from B_j in the implication $\mathbb{C} \vDash All^+(B_i)$ 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 (8.3) 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 $All^+(B_i)$ to false, which is a contradiction. With some more work, and using Theorem 8.2, one can show that $|\mathbb{C}| > N$ if variables from distinct blobs are mixed.

Note 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 8.4. 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 and $W = S$ so that the implication

$$\mathbb{C} \cup \mathbb{B}(W) \vDash All^+(B) \tag{8.5}$$

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

Since by definition $B \cap W = \emptyset$ we have $Vars(All^+(B)) \cap Vars(\mathbb{B}(W)) = \emptyset$, and Theorem 8.2 yields that $|\mathbb{C} \cup \mathbb{B}(W)| \geq |\mathbb{C}[\![W]\!] \cup \mathbb{B}(W)| > |Vars(\mathbb{B}(W))|$, using the notation from (8.2). This is not quite what we want—we have a lower bound on $|\mathbb{C} \cup \mathbb{B}(W)|$, but what we need is a bound on $|\mathbb{C}|$. But if we observe that $|Vars(\mathbb{B}(W))| = d|W|$ while $|\mathbb{B}(W)| = |W|$, we get that

$$|\mathbb{C}| \geq |Vars(\mathbb{B}(W))| - |\mathbb{B}(W)| + 1 = (d-1)|W| + 1 \geq |W| + 1 \tag{8.6}$$

as claimed.

We remark 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 on 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 first degree pebbling formulas.

In general, matters are more complicated than in Examples 8.3 and 8.4. 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 8.5. *Suppose that G is a blob-pebblable DAG and let $\mathbb{C}$ be a set of clauses derived from the pebbling formula $^*Peb^d_G$ for $d \geq 2$. Then $|\mathbb{C}| \geq \mathit{cost}(\mathbb{S}(\mathbb{C}))$.*

Proof. Suppose that the induced set of blob subconfigurations is $\mathbb{S}(\mathbb{C}) = \left\{ [B_i]\langle W_i \rangle \middle| i \in [m] \right\}$. By Definition 6.9, we have $\mathit{cost}(\mathbb{S}(\mathbb{C})) = \left| \mathcal{B} \cup \mathcal{W}^{\Delta} \right|$ where

$$\mathcal{B} = \left\{ \mathrm{bot}(B_i) \middle| [B_i]\langle W_i \rangle \in \mathbb{S}(\mathbb{C}) \right\} \tag{8.7}$$

and

$$\mathcal{W}^{\Delta} = \bigcup_{[B_i]\langle W_i \rangle \in \mathbb{S}(\mathbb{C})} \left(W_i \cap G_{\Delta}^{\mathrm{bot}(B_i)} \right) \quad . \tag{8.8}$$

We need to prove that $|\mathbb{C}| \geq \left|\mathcal{B} \cup \mathcal{W}^{\Delta}\right|$.

We first show that all vertices in $\mathcal{B} \cup \mathcal{W}^{\Delta}$ are represented in some clause in $\mathbb{C}$. By Definition 7.2, for each $[B_i]\langle W_i \rangle \in \mathbb{S}(\mathbb{C})$ there is a clause set $\mathbb{C}_i \subseteq \mathbb{C}$ and a vertex set $S_i \subseteq G \setminus B_i$ with $W_i = S_i \cap \mathit{lpp}(B_i) \subseteq S_i$ such that

$$\mathbb{C}_i \cup \mathbb{B}(S_i) \vDash \mathit{All}^{+}(B_i) \tag{8.9}$$

and such that this implication does not hold for any strict subset of $\mathbb{C}_i$, S_i or B_i. Fix (arbitrarily) such $\mathbb{C}_i$ and S_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 $\mathrm{bot}(B_i) \in V(\mathbb{C})$. To establish this claim, note that for any $v \in B_i$ we can apply Lemma 8.1 with $D_1 = \bigvee_{j=1}^{d} v_j$ and $D_2 = \mathit{All}^{+}(B_i \setminus \{v\})$ on the implication (8.9), which yields that the vertex v must be represented in $\mathbb{C}_i \cup \mathbb{B}(W_i)$ by some positive literal v_j. Since $B_i \cap S_i = \emptyset$, we have $\mathit{Vars}(\mathbb{B}(S_i)) \cap \mathit{Vars}(\mathit{All}^{+}(B_i)) = \emptyset$ and thus $v_j \in \mathit{Lit}(\mathbb{C}_i)$.

Also, we claim that $S_i \subseteq V(\mathbb{C}_i)$. To see this, note that since $B_i \cap S_i = \emptyset$ and the implication (8.9) is minimal, it follows from Lemma 7.4 that for every $w \in S_i$, all literals $\overline{w}_j$, $j \in [d]$, must be present in $\mathbb{C}_i$. Thus, in particular, it holds that $W_i \cap G_{\Delta}^{\mathrm{bot}(B_i)} \subseteq V(\mathbb{C}_i)$.

We now prove by induction over subsets $R \subseteq \mathcal{B} \cup \mathcal{W}^{\Delta}$ that $|\mathbb{C}[\![R]\!]| \geq |R|$. The theorem clearly follows from this since $|\mathbb{C}| \geq |\mathbb{C}[\![R]\!]|$. (The reader can think of R as 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 $R' \subsetneqq R$. Pick a "topmost" vertex $r \in R$, i.e., such that $G_{r}^{\nabla} \cap R = \emptyset$. We associate a blob subconfiguration $[B_i]\langle W_i \rangle \in \mathbb{S}(\mathbb{C})$ with r as follows. If $r = \mathrm{bot}(B_i)$ 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 $r \in W_i \cap G_{\Delta}^{\mathrm{bot}(B_i)}$, so fix any such subconfiguration. We note that it holds that

$$R \cap G_{\mathrm{bot}(B_i)}^{\nabla} \subseteq \{r\} \tag{8.10}$$

for $[B_i]\langle W_i \rangle$ chosen in this way.

Consider the clause set $\mathbb{C}_i \subseteq \mathbb{C}$ and vertex set $S_i \supseteq W_i$ from (8.9) 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

$$\left|\mathbb{C}_i[\![R]\!]\right| \geq \left|R \cap V(\mathbb{C}_i)\right| \quad . \tag{8.11}$$

Let us first see that this claim is sufficient to prove the theorem. To this end, let

$$R[i] = R \cap V(\mathbb{C}_i) \tag{8.12}$$

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,

$$\mathbb{C}[\![R \setminus R[i]]\!] \subseteq \mathbb{C}[\![R]\!] \setminus \mathbb{C}_i \tag{8.13}$$

in our notation. Combining the (yet unproven) claim (8.11) for $\mathbb{C}_i[\![R]\!] = \mathbb{C}_i[\![R[i]]\!]$ asserting that $\bigl|\mathbb{C}_i[\![R[i]]\!]\bigr| \geq |R[i]|$ with the induction hypothesis for $R \setminus R[i] \subseteq R \setminus \{r\} \subsetneqq R$ we get

$$\begin{aligned}\bigl|\mathbb{C}[\![R]\!]\bigr| &= \bigl|\mathbb{C}_i[\![R]\!] \mathbin{\dot\cup} (\mathbb{C} \setminus \mathbb{C}_i)[\![R]\!]\bigr| \\ &\geq \bigl|\mathbb{C}_i[\![R \cap V(\mathbb{C}_i)]\!] \mathbin{\dot\cup} \mathbb{C}[\![R \setminus V(\mathbb{C}_i)]\!]\bigr| \\ &= \bigl|\mathbb{C}_i[\![R[i]]\!]\bigr| + \bigl|\mathbb{C}[\![R \setminus R[i]]\!]\bigr| \\ &\geq |R[i]| + |R \setminus R[i]| \\ &= |R|\end{aligned} \tag{8.14}$$

and the theorem follows by induction.

It remains to verify the claim (8.11) 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 8.2 on the formula $F = \mathbb{C}_i \cup \mathbb{B}(S_i)$ on the left-hand side of the minimal implication (8.9). Let $R' = R[i] \setminus \{r\}$, write $R' = R_1 \mathbin{\dot\cup} R_2$ for $R_1 = R' \cap S_i$ and $R_2 = R' \setminus R_1$, and consider the subformula

$$\begin{aligned}F_{R'} &= \bigl\{C \in \bigl(\mathbb{C}_i \cup \mathbb{B}(S_i)\bigr) \bigm| V(C) \cap R' \neq \emptyset\bigr\} \\ &= \mathbb{C}_i[\![R']\!] \cup \mathbb{B}(R_1)\end{aligned} \tag{8.15}$$

of $F = \mathbb{C}_i \cup \mathbb{B}(S_i)$. A key observation for the concluding part of the argument is that by (8.10) we have $\mathit{Vars}^d(R') \cap \mathit{Vars}(\mathit{All}^+(B_i)) = \emptyset$.

For each $w \in R_1$, the clauses in $\mathbb{B}(R_1)$ contain d literals $w_1, \ldots, w_d$ and these literals must all occur negated in $\mathbb{C}_i$ by Lemma 7.4. For each $u \in R_2$, the clauses in $\mathbb{C}_i[\![R']\!]$ contain at least one variable u_i. Appealing to Theorem 8.2 with the subset of variables $\mathit{Vars}^d(R') \cap \mathit{Vars}(\mathbb{C}_i) \subseteq \mathit{Vars}(F) \setminus \mathit{Vars}(\mathit{All}^+(B_i))$, we get

$$\begin{aligned}\bigl|F_{R'}\bigr| &= \bigl|\mathbb{C}_i[\![R']\!] \cup \mathbb{B}(R_1)\bigr| \\ &\geq \bigl|\mathit{Vars}^d(R') \cap \mathit{Vars}(\mathbb{C}_i)\bigr| + 1 \\ &\geq d\bigl|R_1\bigr| + \bigl|R_2\bigr| + 1 \ ,\end{aligned} \tag{8.16}$$

and rewriting this as

$$\begin{aligned}\bigl|\mathbb{C}_i[\![R[i]]\!]\bigr| &\geq \bigl|\mathbb{C}_i[\![R']\!]\bigr| \\ &= \bigl|F_{R'}\bigr| - \bigl|\mathbb{B}(R_1)\bigr| \\ &\geq (d-1)\bigl|R_1\bigr| + \bigl|R_2\bigr| + 1 \\ &\geq \bigl|R[i]\bigr|\end{aligned} \tag{8.17}$$

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 (8.17). 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 (8.10) holds. It is thanks to this fact that the inequalities in (8.16) go through. The way we make sure that (8.10) holds is to charge only for (distinct) bottom vertices in the black blobs, and only for supporting white pebbles below these bottom vertices.

9 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 *Blob-Peb*(G_i) for some nice family of blob-pebblable DAGs G_i.

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 [42] for the restricted case of trees. There, for the pebblings $\mathcal{P}_\pi$ that one gets from resolution derivations $\pi : {}^*Peb^d_T \vdash \bigvee_{i=1}^{d} z_i$ 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 asymptotically the same cost. The lower bound on pebbling price in the labelled pebbel game then follows immediately by using the known lower bound for black-white pebbling of trees in Theorem 4.8.

Unfortunately, the blob-pebble game seems more difficult than the game in [42] 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 not hard 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 [42] played on binary trees, we are currently not able 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, for instance, [24, 31, 37, 39] all break down for the more general blob-pebble game. We are currently able to obtain lower bounds only for the limited 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 in [37], 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 [37], in the process simplifying the proof somewhat. In the next section we build on this result to generalize the bound from the black-white pebble game to the blob-pebble game in Definition 6.8.

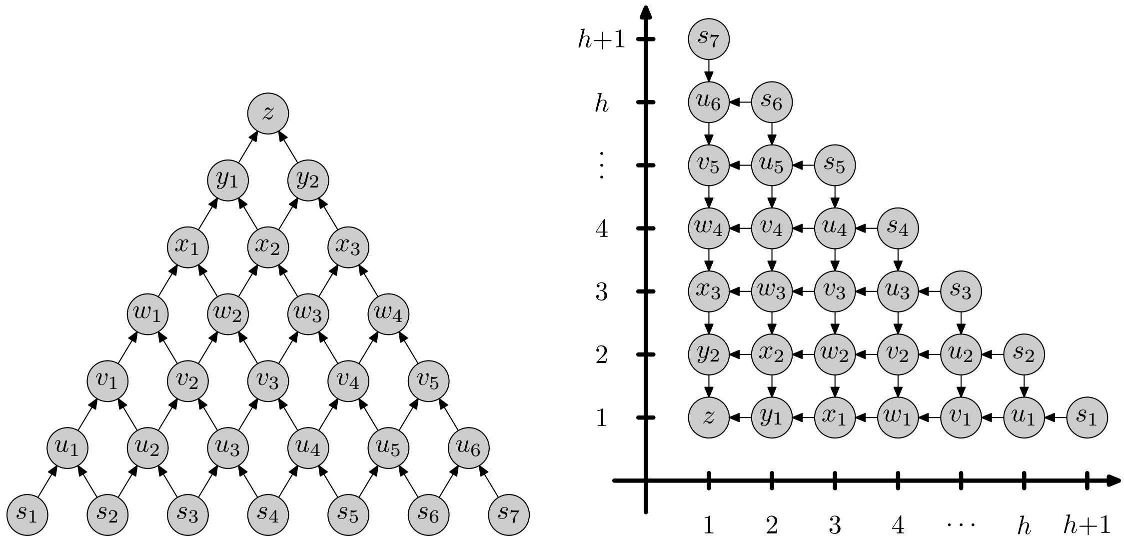
9.1 Some Preliminaries and a Tight Bound for Black Pebbling

Unless otherwise stated, in the following G denotes a layered DAG; u, v, w, x, y denote vertices of G; 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 9.1 (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 $\mathrm{maxlevel}(U) = \max\{\mathrm{level}(u) : u \in U\}$ denote the lowest and highest level, respectively, of any vertex in U. Vertices in U on particular levels are denoted as follows:

- $U\{\succeq j\} = \{u \in U \mid \mathrm{level}(u) \geq j\}$ denotes the subset of all vertices in U on level j or higher.
- $U\{\succ j\} = \{u \in U \mid \mathrm{level}(u) > j\}$ 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 wholly analogously.



(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.

For the layered DAGs G under consideration we will assume that all sources are on level 0, that all non-sources have indegree 2, and that there is a a unique sink z. Since all layered DAGs also possess the Sibling non-reachability property 6.2, this means that we are considering blob-pebblable DAGs (Definition 6.6), and so the blob-pebble game can be played on them.

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.

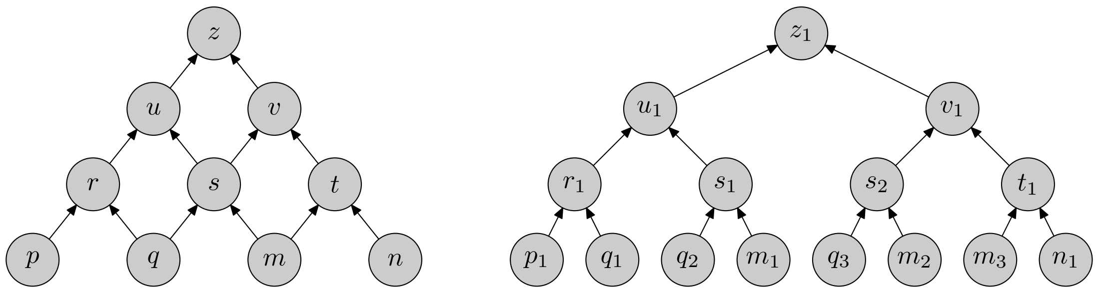
In the standard black and black-white pebble games, we have the following upper bounds on pebbling price of layered DAGs.

Lemma 9.2. *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 $\mathsf{Peb}(G_h) \leq h + \mathrm{O}(1)$ and $\mathsf{BW\text{-}Peb}(G_h) \leq h/2 + \mathrm{O}(1)$.*

Proof. The bounds above are true for complete binary trees of height h according to Theorem 4.8. 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 $\mathit{pred}(w) = \{u, v\}$. Then label the vertices $\mathit{pred}(w_i)$ 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 $\mathit{pred}(w_i) = \{u_j, v_k\}$ in T_h, then $\mathit{pred}(w) = \{u, v\}$ 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. ☐



(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 9.2.

In this section, we will identify some layered graphs G_h for which the bound in Lemma 9.2 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 9.3 ([22]). *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 9.4. A vertex set U *blocks* a path P if $U \cap P \neq \emptyset$. U blocks a set of paths $\mathfrak{P}$ if U blocks all $P \in \mathfrak{P}$.

Proof of Theorem 9.3. It is easy to devise (inductively) a black pebbling strategy that uses $h+2$ pebbles (using, for instance, Lemma 9.2). 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. Suppose that P is (one of) the last path(s) blocked. Obviously, P is 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 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 9.5. *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 $P = \{v_0 = u, v_1, \ldots, v_K = w\}$. Then there is a set of K paths $\mathfrak{P} = \{P_1, \ldots, P_K\}$ such that P_i coincides with P from v_i onwards to w 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 $P_1, \ldots, P_K$ as a set of *converging source paths*, or just converging paths, for $P : u \rightsquigarrow w$. See Figure 9 for an example.

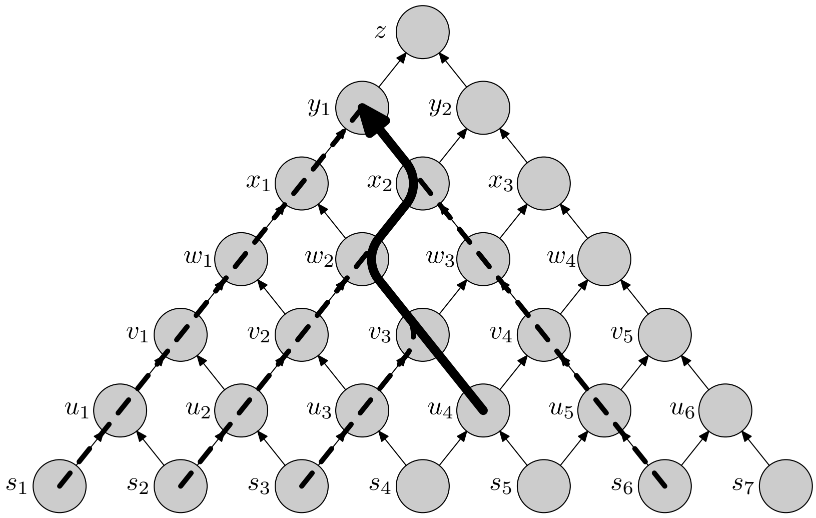


Figure 9: Set of converging source paths (dashed) for the path $P : u_4 \rightsquigarrow y_1$ (solid).

9.2 A Tight Bound on the Black-White Pebbling Price of Pyramids

The rest of this section contains an exposition of Klawe [37], with some simplifications of the proofs. Much of the notation and terminology has been changed from [37] to fit better with this paper in general and (in the next section) the blob-pebble game in particular. Also, it should be noted that we restrict all definitions to layered graphs, in contrast to Klawe who deals with a somewhat more general class of graphs. We concentrate on layered graphs mainly to avoid unnecessary complications in the exposition, and since it can be proven that no graphs in [37] can give a better size/pebbling price trade-off than one gets for layered graphs anyway.

Recall from Definition 6.5 that a *path via* w is a path P such that $w \in P$. We will also say that P *visits* w. The notation $\mathfrak{P}_{\text{via}}(w)$ is used to denote all source paths visiting w. Note that a path $P \in \mathfrak{P}_{\text{via}}(w)$ visiting w may continue after w, or may end in w.

Definition 9.6 (Hiding set). A vertex set U *hides* a vertex w if U blocks all source paths visiting w, i.e., if U blocks $\mathfrak{P}_{\text{via}}(w)$. 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 completely hides w. 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 become clearer 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.

Proposition 9.7. *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 $\mathbb{P} = (B, W)$ 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 defined as the minimum measure 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 9.8 (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 9.9 (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

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

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 9.9 is a relevant estimation of how "good" a pebble configuration is. We refer the reader to Section 2 of [37] 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.

Remark 9.10. Klawe does not use the level of a vertex u in Definitions 9.8 and 9.9, but instead the black pebbling price $\mathsf{Peb}(\{u\}, \emptyset)$ 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 $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ 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, $\text{pot}(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathsf{cost}(\mathbb{P}_s)\}$.
2. The final configuration $\mathbb{P}_\tau = (\{z\}, \emptyset)$ has high potential, namely $\text{pot}(\{z\}, \emptyset) = h + \text{O}(1)$.

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 9.11 (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 9.11, the induction proof for part 1 follows quite easily. The main part of the paper [37] is then spent on proving that a class of DAGs including pyramids have Property 9.11. Let us see what the lower bound proof looks like, assuming that Property 9.11 holds.

Lemma 9.12 (Theorem 2.2 in [37]). *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 9.11, 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

$$\mathrm{pot}(\mathbb{P}_{t+1}) \leq \max\{\mathrm{pot}(\mathbb{P}_t), 2 \cdot \mathit{cost}(\mathbb{P}_{t+1})\} \ . \tag{9.1}$$

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 9.7. 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 $pred(v) \subseteq B_t \cup W_t$. 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 $\mathrm{pot}(\mathbb{P}_t) \leq 2 \cdot \textit{cost}(\mathbb{P}_t)$ holds, only that $\mathrm{pot}(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\textit{cost}(\mathbb{P}_s)\}$. This means that it can happen that $\mathrm{pot}(\mathbb{P}_{t+1}) > 2 \cdot \textit{cost}(\mathbb{P}_{t+1})$, in which case the induction step fails. However, we claim that using the Limited hiding-cardinality property 9.11 we can prove for $U_{t+1} = U_t \cup \{v\}$ that

$$m(U_{t+1}) = m(U_t \cup \{v\}) \leq \max\{m(U_t), 2 \cdot \textit{cost}(\mathbb{P}_{t+1})\} \ , \tag{9.2}$$

   which shows that (9.1) holds and the induction steps goes through.

   Namely, suppose that U_t is chosen as in Property 9.11 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 $|U_t| \leq \textit{cost}(\mathbb{P}_t)$ holds by the LHC property, we have

$$m^0(U_{t+1}) = 2 \cdot |U_{t+1}| = 2 \cdot (|U_t| + 1) \leq 2 \cdot (\textit{cost}(\mathbb{P}_t) + 1) = 2 \cdot \textit{cost}(\mathbb{P}_{t+1}) \tag{9.3}$$

   and we get that

$$\begin{aligned} m(U_{t+1}) &= \max_j\{m^j(U_{t+1})\} = \max\{\max_{j>0}\{m^j(U_t)\}, m^0(U_{t+1})\} \\ &\leq \max\{m(U_t), 2 \cdot \textit{cost}(\mathbb{P}_{t+1})\} = \max\{\mathrm{pot}(\mathbb{P}_t), 2 \cdot \textit{cost}(\mathbb{P}_{t+1})\} \end{aligned} \tag{9.4}$$

   which is exactly what we need.

We see that the inequality (9.1) 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 9.13. *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 9.5, 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

$$m(U') \geq m^L(U') = L + 2\big|U'\{\succeq L\}\big| = L + 2\big|U'\big| \geq L + 2 \cdot (h + 1 - L) > h + 2 \tag{9.5}$$

(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 [37] 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 9.14 ([37]). *BW-Peb*$(\Pi_h) = \frac{h}{2} + \mathrm{O}(1)$.

Proof. The upper bound was shown in Lemma 9.2. For the lower bound, Lemma 9.13 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 9.12, $\mathrm{pot}(\{z\}, \emptyset) \leq 2 \cdot \mathit{cost}(\mathcal{P})$. Thus *BW-Peb*$(\Pi_h) \geq h/2+1$. □

In the final two subsections 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.

9.3 Proving 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 9.20 and Lemma 9.24.

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 9.25, but note that the argument does not quite work. However, if we assume a slightly stronger property locally for our disjoint components (Property 9.27), the proof goes through.

3. We then prove this stronger local property by assuming that pyramid graphs have a certain *spreading* property (Definition 9.34 and Theorem 9.35), and by showing in Lemmas 9.33 and 9.36 that the stronger local property holds for such spreading graphs.

4. Finally, in Section 9.4, we give a simplified proof of the theorem in [37] that pyramids are indeed spreading.

From this, the desired conclusion follows.

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

Definition 9.15 (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 9.16 (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 $u \in U$, then $U\llbracket u \rrbracket = \{u\}$. This is not the case for non-tight sets. If we let $U = \{u\} \cup \mathit{pred}(u)$ for some non-source u, Definition 9.16 yields that $U\llbracket u \rrbracket = \emptyset$. 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., $x \in \llbracket U\llbracket x \rrbracket \rrbracket$.

Lemma 9.17 (Lemma 3.1 in [37]). *If U is tight and $x \in \llbracket U \rrbracket$, 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 $x \in \llbracket U\llbracket x\rrbracket\rrbracket$. 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_1in 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, $x \in \llbracket U\llbracket x\rrbracket\rrbracket$ must hold. ☐

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

Lemma 9.18. *For any vertex set U in a layered graph G there is a uniquely determined minimal subset $U^* \subseteq U$ such that $\llbracket U^*\rrbracket = \llbracket U\rrbracket$, U^* is tight, and for any $U' \subseteq U$ with $\llbracket U'\rrbracket = \llbracket U\rrbracket$ 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 $L = \mathrm{minlevel}(U)$. 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 $U_L^r = U \setminus \llbracket U_L^*\rrbracket$. 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 $U_i^r = U_{i-1}^r \setminus \llbracket U_i^*\rrbracket$. 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 $M = \mathrm{maxlevel}(U)$. By construction, the invariant

$$\llbracket U_i^*\rrbracket = \llbracket U\{\preceq i\}\rrbracket \tag{9.6}$$

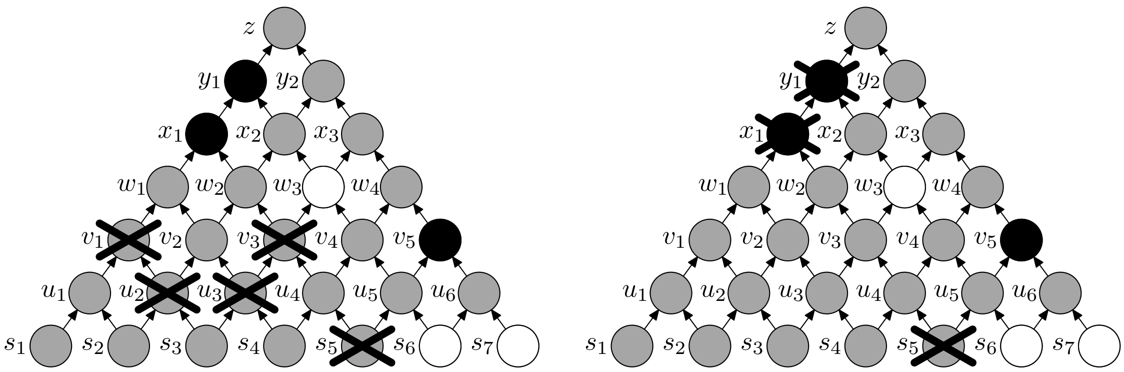
holds for all levels i. Thus, $\llbracket U^*\rrbracket = \llbracket U\rrbracket$. Also, U^* must be tight since if $v \in U^*$ and $\mathrm{level}(v) = i$, by construction $U^*\{\prec i\}$ does not hide v, and (as was argued above) neither does $U^*\{\succeq i\} \setminus \{v\}$. Finally, suppose that $U' \subseteq U$ is a hiding set for U with $U^* \not\subseteq U'$. Consider $v \in U^* \setminus U'$ and suppose $\mathrm{level}(v) = i$. On the one hand, we have $v \notin \llbracket U_{i-1}^*\rrbracket$ by construction. On the other hand, by assumption it holds that $v \in \llbracket U'\{\prec i\}\rrbracket$ and thus $v \in \llbracket U\{\prec i\}\rrbracket$. But then by the invariant (9.6) we know that $v \in \llbracket U_{i-1}^*\rrbracket$, which yields a contradiction. Hence, $U^* \subseteq U'$ and the lemma follows. ☐

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 9.18 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 9.19. 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



(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 9.19 (with vertices in hiding sets cross-marked).

$\{v_1, u_2, u_3, v_3\}$ 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 9.20 (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:

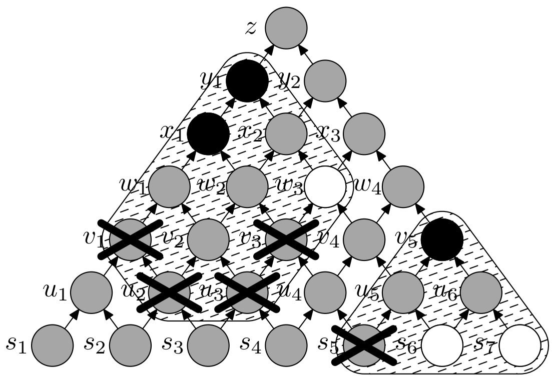
- The set of vertices of $\mathcal{H}$ is $V(\mathcal{H}) = \lceil\!\lceil X \rceil\!\rceil$.
- The set of edges $E(\mathcal{H})$ of $\mathcal{H}$ consists of all pairs of vertices (x, y) for $x, y \in \lceil\!\lceil X \rceil\!\rceil$ such that $G^x_\triangle \cap \lceil\!\lceil X\lfloor\!\lfloor x \rfloor\!\rfloor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

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

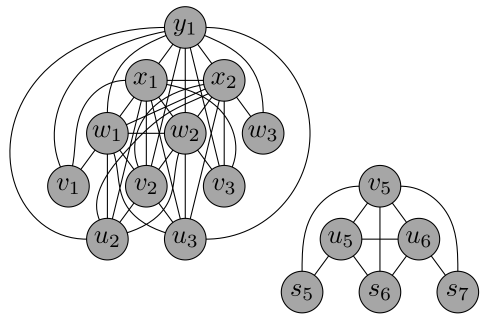
When the graph G and vertex set X are clear from context, we will sometimes write only $\mathcal{H}(X)$ or even just $\mathcal{H}$. To illustrate Definition 9.20, we give an example.

Example 9.21. Consider again the pebble configuration $(B, W) = (\{x_1, y_1, v_5\}, \{w_3, s_6, s_7\})$ from Example 9.19 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 9.19, $\mathcal{H}(X)$ 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 9.22). Perhaps a more interesting edge in $\mathcal{H}(X)$ is, for instance, (w_1, x_2). This edge exists since $X\lfloor\!\lfloor w_1 \rfloor\!\rfloor = \{v_1, u_2, u_3\}$ and $X\lfloor\!\lfloor x_2 \rfloor\!\rfloor = \{u_2, u_3, v_3, w_3\}$ intersect and since as a consequence of this (which is easily verified) we have $\Pi^{w_1}_\triangle \cap \lceil\!\lceil X\lfloor\!\lfloor w_1 \rfloor\!\rfloor \rceil\!\rceil \cap \Pi^{x_2}_\triangle \cap \lceil\!\lceil X\lfloor\!\lfloor x_2 \rfloor\!\rfloor \rceil\!\rceil \neq \emptyset$. For the same reason, there is an edge (u_5, u_6) since $X\lfloor\!\lfloor u_5 \rfloor\!\rfloor = \{s_5, s_6\}$ and $X\lfloor\!\lfloor u_6 \rfloor\!\rfloor = \{s_6, s_7\}$ intersect.



(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.

Lemma 9.22. *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 9.23. *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 $\mathcal{H}(X)$.*

The next lemma says that if $\mathcal{H}(X)$ is a hiding set graph with vertex set $V = \llbracket X \rrbracket$, 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 9.24 (Lemma 3.3 in [37]). *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 $V_i = \llbracket X \cap V_i \rrbracket$.*

Proof. We need to show that $V_i = \llbracket X \cap V_i \rrbracket$ 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 $y \in \llbracket X \cap V_i \rrbracket$.

(⇒) Suppose $y \in V_i$. Then $X\llbracket y \rrbracket \subseteq V_i$ by Corollary 9.23. Also, $X\llbracket y \rrbracket \subseteq X$ by definition, so $X\llbracket y \rrbracket \subseteq X \cap V_i$. Since $y \in \llbracket X\llbracket y \rrbracket \rrbracket$ by Lemma 9.17, clearly $y \in \llbracket X \cap V_i \rrbracket$.

(⇐) Suppose $y \in \llbracket X \cap V_i \rrbracket$. Since X is tight, its subset $X \cap V_i$ must be tight as well. Applying Lemma 9.17 twice, we deduce that $(X \cap V_i)\llbracket y \rrbracket$ hides y and that $X\llbracket y \rrbracket \subseteq (X \cap V_i)\llbracket y \rrbracket$ since $X\llbracket y \rrbracket$ is contained in any subset of X that hides y. But then a third appeal to Lemma 9.17 yields that $(X \cap V_i)\llbracket y \rrbracket \subseteq X\llbracket y \rrbracket$ since $X\llbracket y \rrbracket \subseteq (X \cap V_i)\llbracket y \rrbracket \subseteq X \cap V_i$ and consequently

$$X\llbracket y \rrbracket = (X \cap V_i)\llbracket y \rrbracket \quad . \tag{9.7}$$

By Corollary 9.23, y and all of $(X \cap V_i)\llbracket y \rrbracket = X\llbracket y \rrbracket$ are in the same connected component. Since $X\llbracket y \rrbracket \subseteq V_i$ it follows that $y \in V_i$.

This shows that $V_i = \llbracket X \cap V_i \rrbracket$. Plugging (9.7) into Definition 9.20, 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 9.25 (Analogue of Theorem 3.7 in [37]). *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 9.25. Let U be a set of vertices in 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 9.18) 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 9.24 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 9.11 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 $m\left(U_i^*\right) \leq m(U_i)$. 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 $m\left(U_i^*\right) \leq m\left(U_i\right)$ should imply $m\left((U \setminus U_i) \cup U_i^*\right) \leq m\left((U \setminus U_i) \cup U_i\right)$. And this turns out not to be true. □

The reason that the proof above does not quite work is that the measure in Definition 9.8 is ill-behaved with respect to unions. Klawe provides the following example of what can happen.

Example 9.26. 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 9.27 (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 $m\left(Y \cup U^*\right) \leq m(Y \cup U)$,
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 $m\big(U^*\big) \leq m(U)$ (just choose $Y = \emptyset$). Also, we immediately get that the LHC property holds globally.

Lemma 9.28. *If G has the Local limited hiding-cardinality property 9.27, then G has the Limited hiding-cardinality property 9.11.*

Proof. Consider the tentative proof of Theorem 9.25 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 $m\big(U_i^*\big) \leq m\big(U_i\big)$ does indeed imply $m\big((U \setminus U_i) \cup U_i^*\big) \leq m\big((U \setminus U_i) \cup U_i\big)$, and the theorem follows. ☐

An obvious way to get the inequality $m(Y \cup U^*) \leq m(Y \cup U)$ in Property 9.27 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 9.8.

Definition 9.29. 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 9.29 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. 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 9.26, $X_1 \not\precsim_m X_2$ 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 9.26 can no longer occur.

Lemma 9.30 (Lemma 3.4 in [37]). *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

$$\begin{aligned} m^j(Y \cup U) = j + 2 \cdot |(U \cup Y)\{\succeq j\}| \leq j + 2 \cdot |U\{\succeq j\}| + 2 \cdot |Y\{\succeq j\}| \leq \\ i + 2 \cdot |V\{\succeq i\}| + 2 \cdot |Y\{\succeq i\}| = m^i(Y \cup V) \end{aligned} \tag{9.8}$$

and the lemma follows. ☐

So when locally improving a blocking 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 9.29 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 9.31. 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 9.32. *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 9.31.

Lemma 9.33 (Lemma 3.5 in [37]). *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

$$\left|V^*\right| = |V\{\prec k\}| + |V'| < |V\{\prec k\}| + |V\{\succeq k\}| = |V| \ . \tag{9.9}$$

Combining (9.9) with part 1 of Observation 9.32, 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 (9.9) 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 9.34 (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*

$$|X| \geq L_{\succeq j}(\llbracket X \rrbracket) + j - \mathrm{minlevel}(X) \tag{9.10}$$

holds.

Let us try to give some more intuition for Definition 9.34 by considering two extreme cases in a pyramid graph:

- For $j \leq \mathrm{minlevel}(X)$, we have that the term $j - \mathrm{minlevel}(X)$ is non-positive, $X\{\succeq j\} = X$, and $\llbracket X\{\succeq j\} \rrbracket = \llbracket X \rrbracket$. In this case, (9.10) is just the trivial fact that no set that hides $\llbracket X \rrbracket$ need be larger than X itself.
- Consider $j = \mathrm{maxlevel}(\llbracket X \rrbracket)$, and suppose that $\llbracket X\{\succeq j\} \rrbracket$ is a single vertex v with $X_{\llbracket x \rrbracket} = X$. Then (9.10) requires that $|X| \geq 1 + \mathrm{level}(x) - \mathrm{minlevel}(X)$, and this can be proven to hold by the "converging paths" argument of Theorem 9.3 and Observation 9.5.

Very loosely, Definition 9.34 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 (9.10) does indeed hold for pyramids.

Theorem 9.35 ([37]). *Pyramids are spreading graphs.*

Unfortunately, the proof of Theorem 9.35 in [37] 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. In Section 9.4, we give a simplified, direct proof of the fact that pyramids are spreading that might be of independent interest.

Accepting Theorem 9.35 on faith for now, 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 9.27 to hold, then replacing U by the minimum-size hiding set in Lemma 9.33 we get a hiding set in accordance with the Local LHC property.

Lemma 9.36 (Lemma 3.6 in [37]). *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 9.36 for a moment, let us note that if we combine this lemma with Lemma 9.30 and Theorem 9.35, the Local limited hiding-cardinality property for pyramids follows.

Corollary 9.37. *Pyramid graphs have the Local limited hiding-cardinality property 9.27.*

Proof of Corollary 9.37. This is more or less immediate, but we write down the details for completeness. Since pyramids are spreading by Theorem 9.35, Lemma 9.36 says that U^* is a hiding set for (B, W) and that $U^* \precsim_m U$. Lemma 9.30 then yields that $m(Y \cup U^*) \leq m(Y \cup U)$ for all Y with $Y \cap U = \emptyset$. Finally, Lemma 9.36 also tells us that $U^* \subseteq B$ or $|U^*| < |B| + |W|$, and thus all conditions in Property 9.27 are satisfied. □

Continuing by plugging Corollary 9.37 into Lemma 9.28, we get the global LHC property in Theorem 9.25 on page 55. 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 9.35 and Lemma 9.36. We attend to Lemma 9.36 right away, deferring a proof of Theorem 9.35 to the next subsection.

Proof of Lemma 9.36. 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 9.33. 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 9.19 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 9.33 and 9.36 in action.

Suppose first that $j \leq \mathrm{minlevel}(U \cup W) \leq \mathrm{minlevel}(U)$. Then the measure 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 9.19). 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 $m^j\bigl(U^*\bigr)$, 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 $m^L\bigl(U\bigr)$.

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

$$\begin{aligned}
m^j\bigl(U^*\bigr) &= j + 2 \cdot \bigl|U^*\{\succeq j\}\bigr| && \bigl[\text{ by definition of } m^j(\cdot)\ \bigr]\\
&\leq j + 2 \cdot \bigl|U^*\bigr| && \bigl[\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V\ \bigr]\\
&\leq j + 2 \cdot |B| && \bigl[\text{ by construction of } U^* \text{ in Lemma 9.33 }\bigr]\\
&\leq j + 2 \cdot |U| && \bigl[\text{ by assumption } |U| \geq |B| + |W| \geq |B|\ \bigr]\\
&= j + 2 \cdot \bigl|U\{\succeq j\}\bigr| && \bigl[\ U\{\succeq j\} = U \text{ since } j \leq \mathrm{minlevel}(U)\ \bigr]\\
&= m^j(U) && \bigl[\text{ by definition of } m^j(\cdot)\ \bigr]
\end{aligned}$$

   and we can choose $i = j$ in Definition 9.29.

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 $B \subseteq [\![U \cup W]\!]$, so

$$L_{\succeq j}(B) \leq L_{\succeq j}\bigl([\![U \cup W]\!]\bigr) \tag{9.11}$$

   holds by part 2 of Observation 9.32. Moreover, $U \cup W$ is a hiding-connected set of vertices in a spreading graph G, so the spreading inequality in Definition 9.34 says that $|U \cup W| \geq L_{\succeq j}\bigl([\![U \cup W]\!]\bigr) + j - L$, or

$$j + L_{\succeq j}\bigl([\![U \cup W]\!]\bigr) \leq L + |U \cup W| \tag{9.12}$$

   after reordering. Combining (9.11) and (9.12) we have that

$$j + L_{\succeq j}(B) \leq L + |U \cup W| \tag{9.13}$$

   and it follows that

$$\begin{aligned}
m^j(U^*) &= j + 2 \cdot \bigl|U^*\{\succeq j\}\bigr| && \bigl[\text{ by definition of } m^j(\cdot)\ \bigr]\\
&\leq j + \bigl|U^*\{\succeq j\}\bigr| + \bigl|U^*\bigr| && \bigl[\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V\ \bigr]\\
&\leq j + L_{\succeq j}(B) + |B| && \bigl[\text{ by construction of } U^* \text{ in Lemma 9.33 }\bigr]\\
&\leq L + |U \cup W| + |B| && \bigl[\text{ by the inequality (9.13) }\bigr]\\
&\leq L + 2 \cdot |U| && \bigl[\text{ by assumption } |U| \geq |B| + |W|\ \bigr]\\
&= L + 2 \cdot |U\{\succeq L\}| && \bigl[\ U\{\succeq L\} = U \text{ since } L \leq \mathrm{minlevel}(U)\ \bigr]\\
&= m^L(U) && \bigl[\text{ by definition of } m^L(\cdot)\ \bigr]
\end{aligned}$$

   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 9.29.

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

Concluding this subsection, we want to make a comment about Lemmas 9.33 and 9.36 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 9.33 with $U^* = B$ or $\left|U^*\right| < |B|$ and with $\left|U^*\{\succeq j\}\right| \le L_{\succeq j}(B)$ for all j. This will sometimes be a good idea, and sometimes not. Just as in Lemma 9.36, for $j > \operatorname{minlevel}(U \cup W)$ we can always prove that

$$m^j(U^*) \le \operatorname{minlevel}(U \cup W) + |U| + (|B| + |W|) \ . \tag{9.14}$$

The key message of Lemma 9.36 is that replacing U by U^* is a good idea if U is sufficiently large, namely if $|U| \ge |B| + |W|$, in which case we are guaranteed to get $m^j(U^*) \le m^L(U)$ for $L = \operatorname{minlevel}(U \cup W)$.

9.4 Pyramids Are Spreading Graphs

The fact that pyramids are spreading graphs, that is, that they satisfy the inequality (9.10), is a consequence of the following lemma.

Lemma 9.38 (Ice-Cream Cone Lemma). *If X is a tight vertex set in a pyramid Π such that $\mathcal{H}(X)$ is a connected graph with vertex set $V = \llbracket X \rrbracket$, then there is a unique vertex $x \in V$ such that $X = X\llbracket x \rrbracket$ and $V = \llbracket X\llbracket x \rrbracket \rrbracket \subseteq \Pi^x_\Delta$.*

What the lemma says it that for any tight vertex set X, the connected components $V_1, \ldots, V_k$ look like ragged ice-cream cones turned upside down. Moreover, for each "ice-cream cone" V_i, all vertices in $X \cap V_i$ are needed to hide the top vertex. The two connected components in Figure 11 are both examples of such "ice-cream cones."

Before proving Lemma 9.38, we show how this lemma can be used to establish that pyramid graphs are spreading by a converging-paths argument as in Observation 9.5.

Proof of Theorem 9.35. Suppose that X is a tight and hiding-connected set, i.e., such that $\mathcal{H}(X)$ is a single connected component with set of vertices $V = \llbracket X \rrbracket$. Let $x \in V$ be the vertex given by Lemma 9.38 such that $X = X\llbracket x \rrbracket$ and $V = \llbracket X\llbracket x \rrbracket \rrbracket \subseteq \Pi^x_\Delta$, and let $M = \operatorname{level}(x)$.

For any $j \le M$ we have

$$L_{\succeq j}(\llbracket X \rrbracket) \le M - j + 1 \ . \tag{9.15}$$

This is so since there are only so many vertices on level j in Π^x_Δ and the set of all these vertices must hide everything in $\llbracket X \rrbracket$ above level j since $\llbracket X \rrbracket \subseteq \Pi^x_\Delta$.

By assumption X is tight and all of X is needed to hide x, i.e., $X = X\llbracket x \rrbracket$. Pick a vertex $v \in X$ on bottom level $L = \operatorname{minlevel}(X)$. Since $v \in X\llbracket x \rrbracket$ there is a path $P : v \rightsquigarrow x$ such that $P \cap X = \{v\}$. Consider the set of converging source paths for P in Observation 9.5. All these converging paths $P_1, P_2, \ldots, P_{M-L}$ must be blocked by distinct vertices in $X \setminus \{v\}$, since $P_i \cap P_j \subseteq P \setminus \{v\}$ and $P \setminus \{v\}$ does not intersect X. From this the inequality

$$|X| \ge M - L + 1 \tag{9.16}$$

follows. By combining (9.15) and (9.16), we get that

$$|X| - L_{\succeq j}(\llbracket X \rrbracket) \ge M - L + 1 - (M - j + 1) = j - L \tag{9.17}$$

which is the required spreading inequality (9.10). □

The rest of this subsection is devoted to proving the Ice-Cream Cone Lemma. We will use that fact that pyramids are planar graphs where we can talk about left and right. More precisely, the following (immediate) observation will be central in our proof.

Observation 9.39. *Suppose for a planar DAG G that we have a source path P to a vertex w and two vertices $u, v \in G^w_\Delta$ on opposite sides of P. Then any path $Q : u \rightsquigarrow v$ must intersect P.*

Given a vertex v in a pyramid Π, there is a unique path that passes through v and in every vertex u moves to the right-hand successor of u. We will refer to this path as the *north-east path* through v, or just the *NE-path* through v for short, and denote it by $P_{\mathrm{NE}}(v)$. The path through v always moving to the left is the *north-west path* or *NW-path* through v, and is denoted $P_{\mathrm{NW}}(v)$. For instance, for the vertex v_4 in our running example pyramid in Figure 7 we have $P_{\mathrm{NE}}(v_4) = \{s_4, u_4, v_4, w_4\}$ and $P_{\mathrm{NW}}(v_4) = \{s_6, u_5, v_4, w_3, x_2, y_1\}$. To simplify the proofs in what follows, we make a couple of observations.

Observation 9.40. *Suppose that X is a tight set of vertices in a pyramid Π and that $v \in \lceil\!\lceil X \rceil\!\rceil$. Then $\lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil \subseteq \Pi^v_\Delta$.*

Proof. Since all vertices in $X\llbracket v \rrbracket$ have a path to v by definition, it holds that $X\llbracket v \rrbracket \subseteq \Pi^v_\Delta$. Any vertex $u \in \Pi \setminus \Pi^v_\Delta$ must lie either to the left of $P_{\mathrm{NE}}(v)$ or to the right of $P_{\mathrm{NW}}(v)$ (or both). In the first case, $P_{\mathrm{NE}}(u)$ is a path via u that does not intersect $X\llbracket v \rrbracket$, so $u \notin \lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil$. In the second case, we can draw the same conclusion by looking at $P_{\mathrm{NW}}(u)$. Thus, $\left(\Pi \setminus \Pi^v_\Delta\right) \cap \lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil = \emptyset$. ☐

Observation 9.41. *Suppose that X is a tight set of vertices in a DAG G and that $v \in \lceil\!\lceil X \rceil\!\rceil$. Then there is a source path P to v such that $|P \cap X| = 1$.*

Proof. Let P_1 be any source path to v and note that P_1 intersects X since $v \in \lceil\!\lceil X \rceil\!\rceil$. Let y be the last vertex on P_1 in $P_1 \cap X$, i.e., the vertex on the highest level in this intersection. Since X is tight, there is a source path P_2 to y that does not intersect $X \setminus \{y\}$. Let P be the path that starts like P_2 and then switches to P_1 in y. Then $|P \cap X| = |\{y\}| = 1$. ☐

Using Observations 9.40 and 9.41, we can simplify the definition of the hiding set graph. Note that Observation 9.40 is not true for arbitrary layered DAGs, however, or even for arbitrary layered planar DAGs, so the simplification below does not work in general.

Proposition 9.42. *Let $\mathcal{H} = \mathcal{H}(\Pi, X)$ be the hiding set graph for a tight set of vertices X in a pyramid Π, and suppose that $u, v \in \lceil\!\lceil X \rceil\!\rceil$. Then the following conditions are equivalent:*

1. *(u, v) is an edge in $\mathcal{H}$, i.e., $\Pi^u_\Delta \cap \lceil\!\lceil X\llbracket u \rrbracket \rceil\!\rceil \cap \Pi^v_\Delta \cap \lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil \neq \emptyset$.*
2. *$\lceil\!\lceil X\llbracket u \rrbracket \rceil\!\rceil \cap \lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil \neq \emptyset$.*
3. *$X\llbracket u \rrbracket \cap X\llbracket v \rrbracket \neq \emptyset$.*

Proof. The directions (1) ⇒ (2) and (3) ⇒ (2) are immediate. The implication (2) ⇒ (1) also follows easily, since $\lceil\!\lceil X\llbracket u \rrbracket \rceil\!\rceil \subseteq \Pi^u_\Delta$ and $\lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil \subseteq \Pi^v_\Delta$ by Observation 9.40. To prove (2) ⇒ (3), fix some vertex $w \in \lceil\!\lceil X\llbracket u \rrbracket \rceil\!\rceil \cap \lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil$ and let P be a source path to w as in Observation 9.41 with $P \cap X = \{y\}$ for some vertex y. Since $P \cap X\llbracket u \rrbracket \neq \emptyset \neq P \cap X\llbracket u \rrbracket$ by assumption, we have $y \in X\llbracket u \rrbracket \cap X\llbracket v \rrbracket \neq \emptyset$. ☐

As the first part of the proof of Lemma 9.38, we show that all vertices hidden by a hiding-connected set X are contained in a subpyramid, the top vertex of which is also hidden by X. This gives the ice-cream cone shape alluded to by the name of the lemma.

Lemma 9.43. *Let $\mathcal{H} = \mathcal{H}(\Pi, X)$ be the hiding set graph of a hiding-connected vertex set X in a pyramid Π. Then there is a unique vertex $x \in \lceil\!\lceil X \rceil\!\rceil$ such that $\lceil\!\lceil X \rceil\!\rceil \subseteq \Pi^x_\Delta$.*

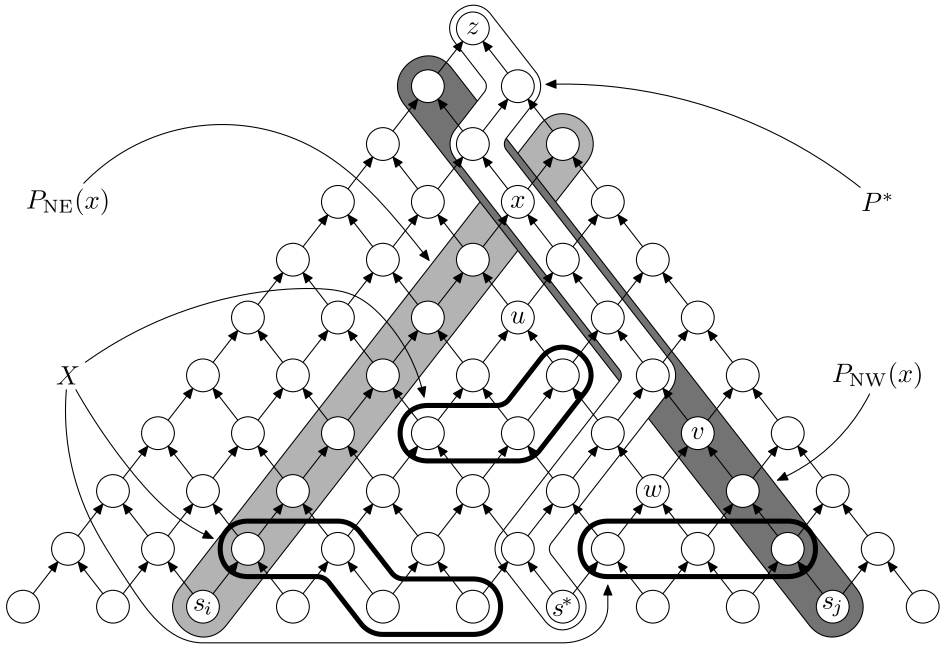


Figure 12: Illustration of proof of Lemma 9.43 that $\mathcal{H}$ is not connected if $x \notin \llbracket X \rrbracket$.

Proof. It is clear that at most one vertex $x \in \llbracket X \rrbracket$ can have the properties stated in the lemma. We show that such a vertex exists. As a quick preview of the proof, we note that it is easy to find a unique vertex x on minimal level such that $\llbracket X \rrbracket \subseteq \Pi^x_\triangle$. The crucial part of the lemma is that x is hidden by X. The reason that this holds is that the graph $\mathcal{H}$ is connected. If $x \notin \llbracket X \rrbracket$, we can find a source path P to the top vertex z of the pyramid such that P does not intersect X but there are vertices in $\mathcal{H}$ both to the left and to the right of P. But there is no way we can have an edge crossing P in $\mathcal{H}$, so the hiding set graph cannot be connected after all. Contradiction.

The above paragraph really is the whole proof, but let us also provide the (somewhat tedious) formal details for completeness. To follow the formalization of the argument, the reader might be helped by looking at Figure 12. Suppose that Π has height h and let $s_1, s_2, \ldots, s_{h+1}$ be the sources enumerated from left to right. Look at the north-east paths $P_{\mathrm{NE}}(s_1), P_{\mathrm{NE}}(s_2), \ldots$ and let s_i be the first vertex such that $P_{\mathrm{NE}}(s_i) \cap \llbracket X \rrbracket \neq \emptyset$. Similarly, consider $P_{\mathrm{NW}}(s_{h+1}), P_{\mathrm{NW}}(s_h), \ldots$ and let s_j be the first vertex such that $P_{\mathrm{NW}}(s_j) \cap \llbracket X \rrbracket \neq \emptyset$. It clearly holds that $i \leq j$.

Let x be the unique vertex where $P_{\mathrm{NE}}(s_i)$ and $P_{\mathrm{NW}}(s_j)$ intersect. By construction, we have $\llbracket X \rrbracket \subseteq \Pi^x_\triangle$, since no NE-path to the left of $P_{\mathrm{NE}}(s_i) = P_{\mathrm{NE}}(x)$ intersects $\llbracket X \rrbracket$ and neither does any NW-path to the right of $P_{\mathrm{NW}}(s_j) = P_{\mathrm{NW}}(x)$. We need to show that it also holds that $x \in \llbracket X \rrbracket$.

To derive a contradiction, suppose instead that $x \notin \llbracket X \rrbracket$. By definition, there is a path P from some source s^* to x such that $P \cap \llbracket X \rrbracket = \emptyset$. P cannot coincide with $P_{\mathrm{NE}}(x)$ or $P_{\mathrm{NW}}(x)$ since the latter two paths both intersect $\llbracket X \rrbracket$ by construction. Since $\Pi^\triangledown_x \cap \llbracket X \rrbracket = \emptyset$, we can extend P to a path $P^* : s^* \rightsquigarrow z$ via x having the property that $P^* \cap \llbracket X \rrbracket = \emptyset$ but there are vertices in $\mathcal{H}(X)$ both to the left and to the right of P^*, namely, the non-empty sets $P_{\mathrm{NE}}(x) \cap \llbracket X \rrbracket \cap \Pi^x_\triangle$ and $P_{\mathrm{NW}}(x) \cap \llbracket X \rrbracket \cap \Pi^x_\triangle$. We claim that this implies that $\mathcal{H}$ is not connected. This is a contradiction to the assumptions in the statement of the lemma and it follows that $x \in \llbracket X \rrbracket$ must hold.

To establish the claim, note that if $\mathcal{H}$ is connected, there must exist some edge (u, v) between a vertex u to the left of P^* and a vertex v to the right of P^*. Then Proposition 9.42 says that $\llbracket X_{\llbracket u \rrbracket} \rrbracket \cap \llbracket X_{\llbracket v \rrbracket} \rrbracket \neq \emptyset$. Pick any vertex $w \in \llbracket X_{\llbracket u \rrbracket} \rrbracket \cap \llbracket X_{\llbracket v \rrbracket} \rrbracket$ and assume without loss of generality that w is on the right-hand side of P^*. We prove that such a vertex w cannot exist. See the example vertices labelled u, v and w in Figure 12, which illustrate the fact that $w \notin \llbracket X_{\llbracket u \rrbracket} \rrbracket$ if

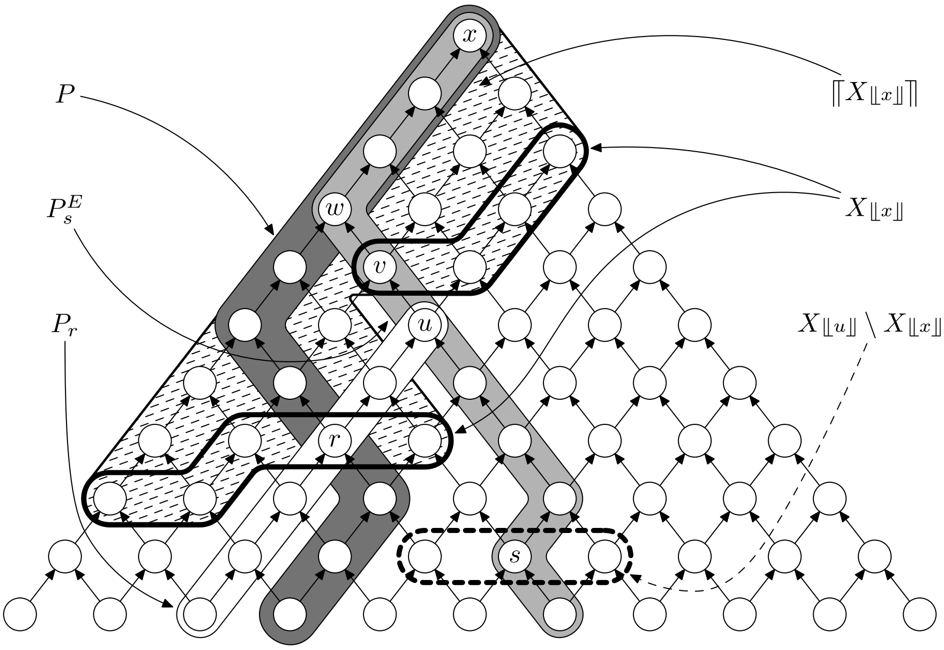


Figure 13: Illustration of proof of Lemma 9.44 that all of X is needed to hide x.

$w \in \lceil\!\lceil X_{\lfloor\!\lfloor v \rfloor\!\rfloor} \rceil\!\rceil$.

Since w is assumed to be hidden by $\lceil\!\lceil X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \rceil\!\rceil$, the NW-path through w must intersect $X_{\lfloor\!\lfloor u \rfloor\!\rfloor}$ somewhere before w or in w. Fix any $y \in P_{\mathrm{NW}}(w) \cap X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \cap \Pi^{w}_{\triangle}$ and note that y must also be located to the right of P^*. By Definition 9.16, there is a source path P' via y to u such that $P' \cap X = \{y\}$. But P' must intersect P^* somewhere above y, since y is to the right and u is to the left of P^*. (Here we use Observation 9.39.) Consider the source path that starts like P^* and then switches to P' at some intersection point in $P' \cap P^* \cap \Pi^{\triangledown}_{y}$. This path reaches u but does not intersect X, contradicting the assumption $u \in \lceil\!\lceil X \rceil\!\rceil$. It follows that $\lceil\!\lceil X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \rceil\!\rceil \cap \lceil\!\lceil X_{\lfloor\!\lfloor v \rfloor\!\rfloor} \rceil\!\rceil = \emptyset$ for all u and v on different sides of P^*, so there are no edges across P^* in $\mathcal{H}$. This proves the claim. ☐

The second part needed to prove Lemma 9.38 is that all vertices in X are required to hide the top vertex $x \in \lceil\!\lceil X \rceil\!\rceil$ found in Lemma 9.43.

Lemma 9.44. *Let $\mathcal{H} = \mathcal{H}(\Pi, X)$ be the hiding set graph of a hiding-connected vertex set X in a pyramid Π and let $x \in \lceil\!\lceil X \rceil\!\rceil$ be the unique vertex such that $\lceil\!\lceil X \rceil\!\rceil \subseteq \Pi^{x}_{\triangle}$. Then $X = X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$.*

Proof. By definition, $X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \subseteq X$. We want to show that $X_{\lfloor\!\lfloor x \rfloor\!\rfloor} = X$. Again, let us first try to convey some intuition why the lemma is true. If $X \setminus X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \neq \emptyset$, since X is hiding-connected there must exist some vertex hidden by all of X but not by just $X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ or $X \setminus X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ (otherwise there can be no edge between the components of $\mathcal{H}$ containing $X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ and $X \setminus X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$, respectively). But if so, it can be shown that the extra vertices in $X \setminus X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ help $X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ to hide one of its own vertices. This contradicts the fact that X is tight, so we must have $X_{\lfloor\!\lfloor x \rfloor\!\rfloor} = X$ which proves the lemma.

Let us fill in the formal details in this proof sketch. Assume, to derive a contradiction, that $X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \neq X$. Since X is tight, it holds that $(X \setminus X_{\lfloor\!\lfloor x \rfloor\!\rfloor}) \cap \lceil\!\lceil X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \rceil\!\rceil = \emptyset$, so $\mathcal{H}$ contains vertices outside of $\lceil\!\lceil X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \rceil\!\rceil$. Since $\mathcal{H}$ is connected, there must exist some edge (u, u') between a pair of vertices $u \in \lceil\!\lceil X \rceil\!\rceil \setminus \lceil\!\lceil X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \rceil\!\rceil$ and $u' \in \lceil\!\lceil X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \rceil\!\rceil$. Lemma 9.17 says that $X_{\lfloor\!\lfloor u' \rfloor\!\rfloor} \subseteq X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ and Proposition 9.42 then tells us that $X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \cap X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \neq \emptyset$. Also, $X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \setminus X_{\lfloor\!\lfloor x \rfloor\!\rfloor} \neq \emptyset$ since $u \notin X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$. For the rest of this proof, fix some arbitrary vertices $r \in X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \cap X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ and $s \in X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \setminus X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$. We refer to Figure 13 for an illustration of the proof from here onwards.

By Definition 9.16, there are source paths P_r via r to u and P_s via s to u that intersect X only in r and s, respectively. Also, there is a source path P to x such that $P \cap X = \{r\}$ since $r \in X\llbracket x \rrbracket$. Suppose without loss of generality that s is to the right of P. The paths P_s and P cannot intersect between s and u. To see this, observe that if P_s crosses P after s but before r, then by starting with P and switching to P_s at the intersection point we get a source path to u that is not blocked by X. And if the crossing is after r, we can start with P_s and then switch to P when the paths intersect, which implies that $s \in X\llbracket x \rrbracket$ contrary to assumption. Thus u is located to the right of P as well.

Extend P_s by going north-west from u until hitting P, which must happen somewhere in between r and x, and then following P to x. Denote this extended path by P_s^E and let w be the vertex starting from which P_s^E and P coincide. The path P_s^E must intersect X in some more vertex after s since $s \notin X\llbracket x \rrbracket$. Pick any $v \in P_s^E \cap (X \setminus \{s\})$. By construction, v must be located strictly between u and w. We claim that $X \setminus \{v\}$ hides v. This contradicts the tightness of X and the lemma follows.

To prove the claim, consider any source path P_v to v and assume that $P_v \cap (X \setminus \{v\}) = \emptyset$. Then, in particular, $r \notin P_v$. Suppose that P_v passes to the left of r. By planarity, P_v must intersect P somewhere above r. But if so, we can construct a source path P' to x that starts like P_v and switches to P at this intersection point. We get $P' \cap X = \emptyset$, which contradicts $x \in X\llbracket x \rrbracket$. If instead P_v passes r on the right, then P_v must cross P_r in order to get to v. This implies that there is a source path P'' to u such that $P'' \cap X = \emptyset$, namely the path obtained by starting to go along P_v and then changing to P_r when the two paths intersect above r. Thus we get a contradiction in this case as well. Hence, $X \setminus \{v\}$ blocks any source path to v as claimed. ☐

The Ice-Cream Cone Lemma 9.38 now follows. Thereby, the proof of the lower bound on the black-white pebbling price of pyramid graphs in Theorem 9.14 on page 51 is complete.

10 A Tight Bound for Blob-Pebbling the Pyramid

Inspired by Klawe's ideas in Section 9, we want to do something similar for the blob-pebble game in Definition 6.8 on page 28. In this section, we study blob-pebblable DAGs (Definition 6.6) that are also layered. We show that for all such DAGs G_h of height h that are spreading in the sense of Definition 9.34, it holds that *Blob-Peb*$(G_h) = \Theta(h)$. In particular, this bound holds for pyramids Π_h since they are spreading by Theorem 9.35.

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. As we have not made any real attempt to get optimal constants, the factor in our lower bound can be improved with a minor effort, but additional ideas seems to be needed to push the constant all the way up to $\frac{1}{2}$.

10.1 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. For a blob B, which is a chain by Definition 6.7, it appears natural to extend this definition by requiring that U should block all paths going through all of B. We recall the terminology and notation from Definition 6.5 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 $\mathfrak{P}_{\text{via}}(B)$ denotes the set of all source paths agreeing with B.

Definition 10.1 (Blocked black blob). A vertex set U *blocks* a blob B if U blocks all $P \in \mathfrak{P}_{\text{via}}(B)$.

A terminological aside: Recalling the discussion in the beginning of Section 9.2, 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 turns out to be a very 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 10.2 (Unblocked paths). For $[B]\langle W\rangle$ an blob subconfiguration, the set of *unblocked paths* for $[B]\langle W\rangle$ is

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

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 9.2, note that when blocking a path $P \in \mathfrak{P}_{\text{via}}(B)$, 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 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 9.8 on page 48, we generalize the concept of *potential* to blob-pebbling configurations as follows.

Definition 10.3 (Blob-pebbling potential). The *potential* of an 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 9.9, consider the following examples with vertex labels as in Figures 7 and 9–11.

Example 10.4. For the blob-pebbling configuration $\mathbb{S} = \{[z]\langle y_1\rangle, [z]\langle y_2\rangle\}$, the minimum-measure blocker 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 10.5. For the blob-pebbling configuration $\mathbb{S} = \{[z]\langle\emptyset\rangle, [y_1]\langle x_1, x_2\rangle\}$, 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 10.6. Perhaps it is also worth pointing out that Definition 10.3 is indeed a strict generalization of Definition 9.9. 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

$$\mathbb{S}(\mathbb{P}) = \big\{ [b] \big\langle W \cap G^b_\vartriangle \big\rangle \big| b \in B \big\} \tag{10.1}$$

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 $\mathbb{S} = \{[z]\langle w_3, w_4\rangle\}$ and the two blocking sets $U_1 = \{z\}$ and $U_2 = \{w_1, w_2\}$.

10.2 A Lower Bound Assuming a Generalized LHC Property

For the blob-pebble game, a useful generalization of Property 9.11 on page 48 turns out to be the following.

Property 10.7 (Generalized limited hiding-cardinality property). We say that a blob-pebbling configuration $\mathbb{S}$ on a layered blob-pebblable DAG G has the *Generalized limited hiding-cardinality property with parameter* C_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. $|U| \leq C_K \cdot \textit{cost}(\mathbb{S})$.

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

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

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

For all layered blob-pebblable DAGs G_h of height h that have the Generalized LHC property and are spreading, it holds that $\textit{Blob-Peb}(G_h) = \Theta(h)$. The proof of this fact is very much in the spirit of the proofs of Lemma 9.12 and Theorem 9.14, although the details are slightly more complicated.

Theorem 10.8 (Analogue of Theorem 9.14). *Suppose that G_h is a layered blob-pebblable DAG of height h possessing the Generalized LHC property 10.7 with some fixed parameter C_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*

$$\mathrm{pot}(\mathbb{S}_t) \leq (2C_K + 1) \cdot \max_{s \leq t}\{\textit{cost}(\mathbb{S}_s)\} \ . \tag{10.2}$$

In particular, for any family of layered blob-pebblable DAGs G_h that are also spreading in the sense of Definition 9.34, we have $\textit{Blob-Peb}(G_h) = \Theta(h)$.

We make two separate observations before presenting the proof.

Observation 10.9. *For any layered DAG G_h of height h it holds that $\textit{Blob-Peb}(G_h) = \mathrm{O}(h)$.*

Proof. Any layered DAG G_h can be black-pebbled with $h + \mathrm{O}(1)$ pebbles by Theorem 9.2 on page 45, and it is easy to see that a blob-pebbling can mimic a black pebbling in the same cost. ☐

Observation 10.10. *If G_h is a layered blob-pebblable DAG of height h that is spreading in the sense of Definition 9.34, 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 9.13. Note, though, that in contrast to Lemma 9.13, 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 9.15 and that $U' = U'\llbracket z \rrbracket$. Then by Corollary 9.23 it holds that U' is hiding-connected. Letting $L = \mathrm{minlevel}\bigl(U'\bigr)$ and setting $j = h$ in the spreading inequality (9.10), we get that $\bigl|U'\bigr| \geq 1 + h - L$ and hence $m\bigl(U'\bigr) \geq m^L\bigl(U'\bigr) \geq L + 2(1 + h - L) = 2h - L + 2 > h + 2$ since $L < h$. □

Proof of Theorem 10.8. The statement in the theorem follows from Observations 10.9 and 10.10 combined with the inequality (10.2), so just as for Theorem 9.14 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 $\mathrm{pot}(\mathbb{S}_t) = m(U_t)$. By the inductive hypothesis, we have that $\mathrm{pot}(\mathbb{S}_t) \leq (2C_K + 1) \cdot \max_{s \leq t}\{\mathit{cost}(\mathbb{S}_s)\}$. We want to show for $\mathbb{S}_{t+1}$ that $\mathrm{pot}(\mathbb{S}_{t+1}) \leq (2C_K + 1) \cdot \max_{s \leq t+1}\{\mathit{cost}(\mathbb{S}_s)\}$. Clearly, this follows if we can prove that

$$\mathrm{pot}(\mathbb{S}_{t+1}) \leq \max\{\mathrm{pot}(\mathbb{S}_t), (2C_K + 1) \cdot \mathit{cost}(\mathbb{S}_t)\} \ . \tag{10.3}$$

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

We make a case analysis depending on the type of move in Definition 6.8 made to get from $\mathbb{S}_t$ to $\mathbb{S}_{t+1}$. Analogously with the proof of Lemma 9.12, we want to show that we can use U_t to block $\mathbb{S}_{t+1}$ as long as the move is not an introduction on a source vertex and then use the Generalized LHC property to take care of such black pebble placements on sources.

Erasure $\mathbb{S}_{t+1} = \mathbb{S}_t \setminus \bigl\{[B]\langle W\rangle\bigr\}$ for $[B]\langle W\rangle \in \mathbb{S}_t$. Obviously, U_t blocks $\mathbb{S}_{t+1} \subseteq \mathbb{S}_t$.

Inflation $\mathbb{S}_{t+1} = \mathbb{S}_t \cup \bigl\{[B]\langle W\rangle\bigr\}$ for $[B]\langle W\rangle$ inflated from some $[B']\langle W'\rangle \in \mathbb{S}_t$ such that

$$B' \subseteq B \ , \tag{10.4a}$$

$$W' \cap \mathit{lpp}(B) \subseteq W \ , \text{ and} \tag{10.4b}$$

$$B \cap W' = \emptyset \ . \tag{10.4c}$$

We claim that U_t blocks $[B]\langle W\rangle$ and thus all of $\mathbb{S}_{t+1}$. Let us first argue intuitively why. Suppose that P is any source path agreeing with B. This path also agrees with B', and so must be blocked by $U_t \cup W'$ by assumption. If U_t blocks B we are done. We can worry, though, that U_t does not block P, but that instead P was blocked by some $w \in W'$ that disappeared as a result of the inflation move. But if $w \in W'$ is on a path via B, it cannot have disappeared, so this can never happen.

We now write down the formal details. With the notation in Definition 10.2, fix any path $P \in \mathrm{unblocked}([B]\langle W\rangle)$. We need to show that $P \cap U_t \neq \emptyset$. Let us assume without loss of generality that P ends in $\mathrm{top}(B)$, for U_t blocks $[B]\langle W\rangle$ precisely if it blocks the paths $P \cap G_{\Delta}^{\mathrm{top}(B)}$ for all $P \in \mathrm{unblocked}([B]\langle W\rangle)$. We note that by definition, the fact that P agrees with a chain V and ends in $\mathrm{top}(V)$ implies that

$$P \subseteq V \mathbin{\dot\cup} \mathit{lpp}(V) \ . \tag{10.5}$$

Since P agrees with B, or in formal notation $P \in \mathfrak{P}_{\text{via}}(B)$, and since $B' \subseteq B$ by (10.4a), we have $P \in \mathfrak{P}_{\text{via}}(B')$. By assumption, U_t blocks $[B']\langle W'\rangle$, which in particular means that

$U_t \cup W'$ intersects the path P agreeing with B'. We get

$$\begin{aligned}
\emptyset \neq P \cap \big(U_t \cup W'\big) & \qquad \big[\text{ by definition of blocking } \big] \\
= (P \cap U_t) \cup \big((P \setminus B) \cap W'\big) & \qquad \big[\text{ since } B \cap W' = \emptyset \text{ by (10.4c) } \big] \\
= (P \cap U_t) \cup \big(P \cap \mathit{lpp}(B) \cap W'\big) & \qquad \big[\text{ since } P \subseteq B \,\dot\cup\, \mathit{lpp}(B) \text{ by (10.5) } \big] \\
\subseteq (P \cap U_t) \cup (P \cap W) & \qquad \big[\text{ since } \mathit{lpp}(B) \cap W' \subseteq W \text{ by (10.4b) } \big] \\
= P \cap U_t & \qquad \big[\ P \cap W = \emptyset \text{ if } P \in \mathrm{unblocked}([B]\langle W\rangle) \ \big]
\end{aligned}$$

so $P \cap U_t \neq \emptyset$ and the desired conclusion that U_t blocks the path P follows.

Merger $\mathbb{S}_{t+1} = \mathbb{S}_t \cup \big\{[B]\langle W\rangle\big\}$ for $[B]\langle W\rangle$ derived by merger of $[B_1]\langle W_1\rangle, [B_2]\langle W_2\rangle \in \mathbb{S}_t$ such that

$$B_1 \cap W_2 = \emptyset \ , \tag{10.6a}$$
$$B_2 \cap W_1 = \{v^*\} \ , \tag{10.6b}$$
$$B = (B_1 \cup B_2) \setminus \{v^*\} \ , \text{ and} \tag{10.6c}$$
$$W = \big((W_1 \cup W_2) \setminus \{v^*\}\big) \cap \mathit{lpp}(B) \ . \tag{10.6d}$$

Let us again first argue informally that if a set of vertices U_t blocks two subconfigurations $[B_1]\langle W_1\rangle$ and $[B_2]\langle W_2\rangle$, it must also block their merger. 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 $U_t \cup W_2$. If on the other hand P agrees with B but does *not* visit v^*, it is a path via B_1 that in addition does not pass through the white pebble in W_1 eliminated in the merger. This means that $U_t \cup W_1 \setminus \{v^*\}$ must block P. Again, we have to argue that the blocking white vertices do not disappear when we apply the intersection with $\mathit{lpp}(B)$ in (10.6d), but this is straightforward to verify.

So let us show formally that U_t blocks $[B]\langle W\rangle$, i.e., that for any $P \in \mathrm{unblocked}([B]\langle W\rangle)$ it holds that $P \cap U_t \neq \emptyset$. As above, without loss of generality we consider only paths P ending in $\mathrm{top}(B) = \mathrm{top}(B_1 \cup B_2)$. Recall that

$$B_i \cap W_i = \emptyset \tag{10.7}$$

holds for all subconfigurations by definition. We divide the analysis into two subcases.

1. $P \in \mathfrak{P}_{\text{via}}(B_1 \cup B_2) = \mathfrak{P}_{\text{via}}(B \cup \{v^*\})$. If so, in particular it holds that $P \in \mathfrak{P}_{\text{via}}(B_2)$ and since U_t blocks $[B_2]\langle W_2\rangle$ we have

$$\begin{aligned}
\emptyset \neq P \cap \big(U_t \cup W_2\big) & \qquad \big[\text{ by definition of blocking } \big] \\
= (P \cap U_t) \cup \big((P \setminus (B_1 \cup B_2)) \cap W_2\big) & \qquad \big[\text{ by (10.6a) and (10.7) } \big] \\
= (P \cap U_t) \cup \big(P \cap \mathit{lpp}(B_1 \cup B_2) \cap W_2\big) & \qquad \big[\text{ by (10.5) } \big] \\
= (P \cap U_t) \cup \big(P \cap \mathit{lpp}(B \cup v^*) \cap W_2\big) & \qquad \big[\text{ just rewriting using (10.6c) } \big] \\
\subseteq (P \cap U_t) \cup \big(P \cap (W_2 \setminus \{v^*\}) \cap \mathit{lpp}(B) & \qquad \big[\ \mathit{lpp}(B \cup \{v^*\}) \subseteq \mathit{lpp}(B) \setminus \{v^*\} \ \big] \\
\subseteq (P \cap U_t) \cup (P \cap W) & \qquad \big[\text{ by (10.6d) } \big] \\
= P \cap U_t & \qquad \big[\text{ since } P \in \mathrm{unblocked}([B]\langle W\rangle) \ \big]
\end{aligned}$$

   so U_t blocks the path P in this case.

2. $P \in \mathfrak{P}_{\text{via}}(B) \setminus \mathfrak{P}_{\text{via}}(B \cup \{v^*\})$. This means that $B \subseteq P$ but $B \cup \{v^*\} \nsubseteq P$, so the path P does not pass through v^*. Since P agrees with B_1 and U_t blocks $[B_1]\langle W_1\rangle$ by

assumption, we get that

$$\begin{aligned}
\emptyset \neq P \cap \bigl(U_t \cup W_1\bigr) & \qquad \bigl[\text{ by definition of blocking }\bigr] \\
= (P \cap U_t) \cup \bigl((P \setminus B) \cap W_1\bigr) & \qquad \bigl[\text{ by (10.6b) and (10.7) }\bigr] \\
= (P \cap U_t) \cup \bigl(P \cap \mathit{lpp}(B) \cap W_1\bigr) & \qquad \bigl[\, P \subseteq B \mathbin{\dot\cup} \mathit{lpp}(B) \text{ by (10.5) }\bigr] \\
= (P \cap U_t) \cup \bigl(P \cap (W_1 \setminus \{v^*\}) \cap \mathit{lpp}(B)\bigr) & \qquad \bigl[\text{ since } v^* \notin P \text{ by assumption }\bigr] \\
\subseteq (P \cap U_t) \cup (P \cap W) & \qquad \bigl[\text{ by (10.6d) }\bigr] \\
= (P \cap U_t) & \qquad \bigl[\, P \in \mathrm{unblocked}([B]\langle W\rangle) \,\bigr]
\end{aligned}$$

and U_t blocks the path P in this case as well.

Introduction $\mathbb{S}_{t+1} = \mathbb{S}_t \cup \bigl\{[v]\langle \mathit{pred}(v)\rangle\bigr\}$. Clearly, U_t blocks $\mathbb{S}_{t+1}$ if v is a non-source vertex, i.e., if $\mathit{pred}(v) \neq \emptyset$, since U_t blocks $\mathbb{S}_t$ and $[v]\langle \mathit{pred}(v)\rangle$ blocks itself.

Suppose however that v is a source vertex, so that the subconfiguration introduced is $[v]\langle\emptyset\rangle$. As in the proof of Lemma 9.12, 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 $|U_t| \leq C_K \cdot \mathit{cost}(\mathbb{S}_t)$ Generalized LHC property 10.7 we have

$$\begin{aligned}
m^0(U_{t+1}) = 2 \cdot |U_{t+1}| = 2 \cdot (|U_t| + 1) \leq \\
2 \cdot \bigl(C_K \cdot \mathit{cost}(\mathbb{S}_t) + 1\bigr) \leq 2 \cdot \bigl(C_K \cdot \mathit{cost}(\mathbb{S}_{t+1}) + 1\bigr) \leq \\
2 \cdot \bigl(C_K \cdot \mathit{cost}(\mathbb{S}_{t+1}) + \mathit{cost}(\mathbb{S}_{t+1})\bigr) \leq 2(C_K + 1) \cdot \mathit{cost}(\mathbb{S}_{t+1})
\end{aligned} \tag{10.8}$$

and we get that

$$\begin{aligned}
\mathrm{pot}(\mathbb{S}_{t+1}) \leq m(U_{t+1}) \leq \max_j\bigl\{m^j(U_{t+1})\bigr\} \\
\leq \max\bigl\{m(U_t), (2C_K + 1) \cdot \mathit{cost}(\mathbb{S}_{t+1})\bigr\} = \\
\max\bigl\{\mathrm{pot}(\mathbb{S}_t), (2C_K + 1) \cdot \mathit{cost}(\mathbb{S}_{t+1})\bigr\}
\end{aligned} \tag{10.9}$$

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

We see that regardless of the pebbling move made in the transition $\mathbb{S}_t \rightsquigarrow \mathbb{S}_{t+1}$, the inequality (10.3) holds. The theorem follows by the induction principle. ☐

Hence, in order to prove a lower bound on $\mathit{Blob\text{-}Peb}(G_h)$ for layered spreading graphs G_h, it is sufficient to find some constant C_K such that these DAGs can be shown to possess the Generalized LHC property 10.7 with parameter C_K.

10.3 Some Structural Transformations

As we tried to indicate by presenting the small toy blob-pebbling configurations in Examples 10.4 and 10.5, 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 10.7 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.[8] Throughout this subsection, we assume that the parameter C_K is some fixed constant.

We start slowly by taking care of a pretty obvious redundancy. Let us say that the blob sub-configuration $[B]\langle W\rangle$ 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 $[B]\langle W\rangle = [v]\langle pred(v)\rangle$ for a non-source vertex v. The following proposition is immediate.

Proposition 10.11. *For $\mathbb{S}$ any blob-pebbling configuration, let $\mathbb{S}'$ be the blob-pebbling configuration with all self-blockers in $\mathbb{S}$ removed. Then $\mathsf{cost}(\mathbb{S}') \leq \mathsf{cost}(\mathbb{S})$, $\mathrm{pot}(\mathbb{S}') = \mathrm{pot}(\mathbb{S})$ and any blocking set U' for $\mathbb{S}'$ is also a blocking set for $\mathbb{S}$.*

Corollary 10.12. *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 $|U| \leq C_K \cdot \mathsf{cost}(\mathbb{S}') \leq C_K \cdot \mathsf{cost}(\mathbb{S})$. Thus the Generalized LHC property holds for $\mathbb{S}$. ☐

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 10.4, it seems that the white pebbles really do not help at all. One might ask if we could not just throw them away? Perhaps somewhat surprisingly, the answer is yes, and we can capture the intuitive concept of necessary white pebbles and formalize it as follows.

Definition 10.13 (White sharpening). Given $\mathbb{S} = \{[B_i]\langle W_i\rangle\}_{i\in[m]}$, we say that $\mathbb{S}'$ is a *white sharpening* of $\mathbb{S}$ if $\mathbb{S}' = \{[B'_i]\langle W'_i\rangle\}_{i\in[m]}$ 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 10.14. *If $\mathbb{S}'$ is a white sharpening of $\mathbb{S}$ it holds that $\mathsf{cost}(\mathbb{S}') \leq \mathsf{cost}(\mathbb{S})$ 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 6.9. The statement about potential clearly follows from Definition 10.3 since it holds that any blocking set U' for $\mathbb{S}'$ is also a blocking set for $\mathbb{S}$. ☐

[8] Note that we did something similar in Section 9.3 after Lemma 9.18, 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 wholly analogous to the reasoning here, but since matters become more complex we need to be a bit more careful.

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 10.15 (White elimination). For $[B]\langle W\rangle$ a 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

$$\mathcal{W}\text{-elim}(\mathbb{S}, U) = \left\{\mathcal{W}\text{-elim}([B]\langle W\rangle, U) \middle| [B]\langle W\rangle \in \mathbb{S}\right\} \ . \tag{10.10}$$

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 10.16. *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 10.13 and 10.15), it holds by Proposition 10.14 that $\mathrm{pot}(\mathbb{S}') \geq \mathrm{pot}(\mathbb{S})$. Looking at the construction in Definition 10.15, 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. □

Corollary 10.17. *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 10.12 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 10.14 and $\mathrm{pot}(\mathbb{S}^1) = \mathrm{pot}(\mathbb{S})$ by Lemma 10.16.

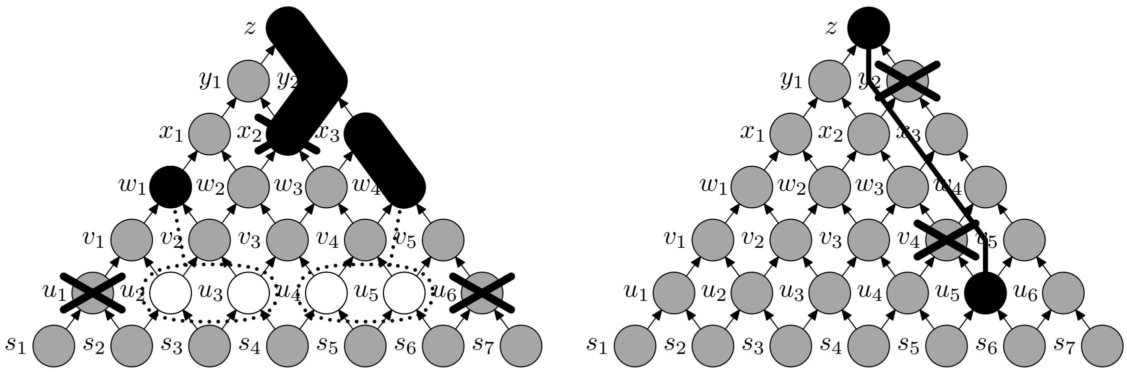
If there is a minimum-measure blocker U^1 for $\mathbb{S}^1$ such that $\mathbb{S}^1$ is not $\mathcal{W}$-eliminated with respect to U^1, set $\mathbb{S}^2 = \mathcal{W}\text{-elim}(\mathbb{S}^1, U^1)$. Continuing in this manner, we get a chain $\mathbb{S}^1, \mathbb{S}^2, \mathbb{S}^3, \ldots$ of strict $\mathcal{W}$-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 $|U^k| \leq C_K \cdot \mathit{cost}(\mathbb{S}^k) \leq C_K \cdot \mathit{cost}(\mathbb{S})$. Thus the Generalized LHC property holds for $\mathbb{S}$. □

We note that in particular, it follows from the construction in Definition 10.15 combined with Corollary 10.17 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 10.18. *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. □



(a) Minimum-measure but non-tight blocking set. (b) Tight but non-connected blocker for blob.

Figure 14: Two blob-pebbling configurations with problematic blocking sets.

10.4 A 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 10.19. *All layered blob-pebblable DAGs that are spreading possess the Generalized limited hiding-cardinality property 10.7 with parameter* $C_K = 13$.

Since pyramids are spreading graphs by Theorem 9.35, this is all that we need to get the lower bound on blob-pebbling price on pyramids from Theorem 10.8. We note that the parameter C_K in Theorem 10.19 can easily be improved. However, our main concern here is not optimality of constants but clarity of exposition.

We prove Theorem 10.19 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 do naive pattern matching on Klawe's proof for the standard black-white pebble game.

In the standard black-white pebble game, if U is a minimum-measure hiding set for $\mathbb{P} = (B, W)$, Lemma 9.18 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 10.3.

Example 10.20. 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 14(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).

This can be handled, but a more serious 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 9.20.

Example 10.21. Consider the single-blob configuration $\mathbb{S} = \{[u_5, z]\langle\emptyset\rangle\}$ in Figure 14(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 9.20 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 9.23 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 9.34. Since the level difference in U is 3 but the size of the set is only 2, the spreading inequality (9.10) 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 9.6 and 10.1, let us make an easy observation relating the hiding and blocking relations for a blob.

Observation 10.22. *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. □

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

Definition 10.23 (Hiding blob-pebbling configurations). We say that the vertex set U *hides* the subconfiguration $[B]\langle W\rangle$ if $U \cup W$ hides the vertex $\mathrm{bot}(B)$, and that U hides 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 10.24. *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 $U \cap G_{\vartriangle}^{\mathrm{bot}(B)}$ 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 $P = \left(P_1 \cap G_{\vartriangle}^{b}\right) \cup \left(P_2 \cap G^{\triangledown}_{b}\right)$ be the source path that starts like P_1 and continues like P_2 from b onwards. Clearly,

$$P \cap \left(\left(U \cap G_{\vartriangle}^{b}\right) \cup W\right) = \left(P_1 \cap (U \cup W)\right) \cup \left(P_2 \cap W\right) = \emptyset \tag{10.11}$$

so $U \cap G_{\vartriangle}^{b}$ 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 $P = \left(P_1 \cap G_{\vartriangle}^{b}\right) \cup \left(P_2 \cap G^{\triangledown}_{b}\right)$ 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 $[B]\langle W\rangle$. □

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

$$\mathbb{S}_H(\mathbb{S}, U) = \left\{[B]\langle W\rangle \in \mathbb{S} \,\middle|\, U \text{ hides } [B]\langle W\rangle\right\} \tag{10.12}$$

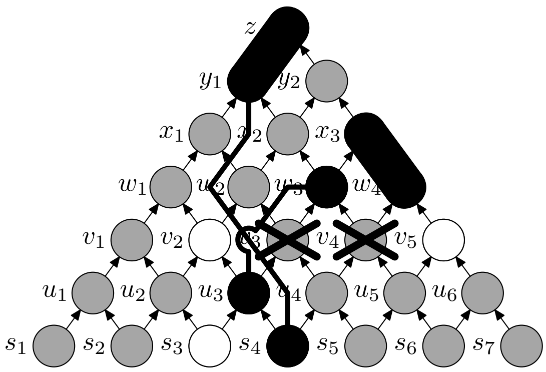
to denote the subconfigurations in $\mathbb{S}$ hidden by U and

$$\mathbb{S}_B(\mathbb{S}, U) = \mathbb{S} \setminus \mathbb{S}_H(\mathbb{S}, U) \tag{10.13}$$

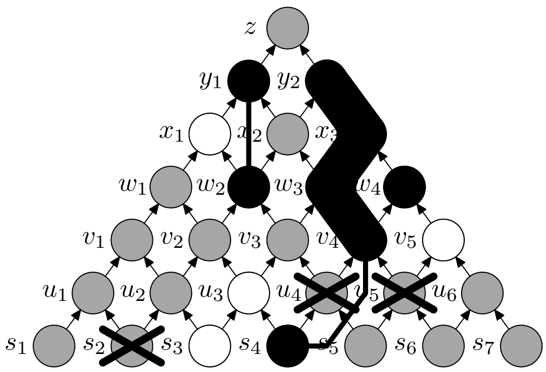
to denote the subconfigurations that are just blocked. We write

$$\mathcal{B}_H(\mathbb{S}, U) = \{\mathrm{bot}(B) \mid [B]\langle W\rangle \in \mathbb{S}_H(\mathbb{S}, U)\} \tag{10.14}$$

$$\mathcal{B}_B(\mathbb{S}, U) = \{\mathrm{bot}(B) \mid [B]\langle W\rangle \in \mathbb{S}_B(\mathbb{S}, U)\} \tag{10.15}$$



(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 15: Examples of blob-pebbling configurations with hidden and just blocked blobs.

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

$$\mathcal{W}_H^{\vartriangle}(\mathbb{S}, U) = \left\{W \cap G_{\vartriangle}^{b} \middle| [B]\langle W\rangle \in \mathbb{S}_H(\mathbb{S}, U),\ b = \mathrm{bot}(B)\right\} \tag{10.16}$$

and

$$\mathcal{W}_B^{\vartriangle}(\mathbb{S}, U) = \left\{W \cap G_{\vartriangle}^{b} \middle| [B]\langle W\rangle \in \mathbb{S}_B(\mathbb{S}, U),\ b = \mathrm{bot}(B)\right\}\ . \tag{10.17}$$

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

Example 10.25. Consider the blob-pebbling configurations and blocking sets in Figure 15. 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 15(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

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

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 15(b), it is straightforward to verify that

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

are the corresponding sets.

Let us also use the opportunity to illustrate Definition 10.15. 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.

As a final remark in this example, we comment that although we have not indicated explicitly in Figures 15(a) and 15(b) which white pebbles W are associated with which black blob B (as was done in Figure 14(a)), this is uniquely determined by the requirement in Definition 6.7 that $W \subseteq lpp(B)$.

For the rest of this section we will assume without loss of generality (in view of Proposition 10.11 and Corollary 10.17) 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 10.15 and Proposition 10.24) 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 10.19 clearly follows if we can prove the following lemma.

Lemma 10.26. *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. $\mathrm{pot}(\mathbb{S}) = m(U)$*, 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* $\mathrm{pot}(\mathbb{S}) = m(U')$.

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

The proof is by contradiction, although we will have to work harder than for the corresponding Theorem 9.25 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 10.26 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 9.3 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 10.21).

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 10.26, and thus Theorem 10.19, 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 level above $\mathrm{level}(\mathrm{bot}(B))$. 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 $\mathcal{B}^u = \{\mathrm{bot}(B) \mid U \setminus \{u\} \text{ does not block } [B]\langle W \rangle \in \mathbb{S}_B\}$. 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 10.27. *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 $\mathit{cost}(\mathbb{S})$, 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 10.28. *Assume that $\mathbb{S}$ and U are as in Lemma 10.26 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 10.27, the proof of the lemma uses the easy observation that moving vertices downwards can only decrease the measure.

Observation 10.29. *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 $f : V_1 \mapsto V_2$ 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 9.8 on page 48 since the mapping f tells us that

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

for all j. ☐

Proof of Lemma 10.27. Note first that by Proposition 10.24, for every $[B]\langle W\rangle \in \mathbb{S}_B$ with $b = \mathrm{bot}(B)$ it holds that $U \cap G_b^{\triangledown} = (U_H \mathbin{\dot\cup} U_B) \cap G_b^{\triangledown}$ 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 $U_H \mathbin{\dot\cup} \left(U_B \cap \bigcup_{b \in \mathcal{B}_B} G_b^{\triangledown}\right)$, and since U is subset-minimal we get that

$$U_B = U_B \cap \textstyle\bigcup_{b \in \mathcal{B}_B} G_b^{\triangledown} \ . \tag{10.18}$$

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 (10.18). 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 $\mathcal{B}' \subsetneqq \mathcal{B}_B$ 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 $\mathrm{level}(u) \geq \mathrm{level}(b)$.

Clearly, all subconfigurations in

$$\mathbb{S}_B^1 = \left\{[B]\langle W\rangle \in \mathbb{S}_B \,\middle|\, \mathrm{bot}(B) \in \mathcal{B}_B \setminus \mathcal{B}'\right\} \tag{10.19}$$

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 $b = \mathrm{bot}(B)$ is blocked by $U_H \cup \bigl(U_B \cap G_b^{\triangledown}\bigr) = U_H \cup N(b)$, so all subconfigurations in

$$\mathbb{S}_B^2 = \bigl\{[B]\langle W\rangle \in \mathbb{S}_B \bigm| \mathrm{bot}(B) \in \mathcal{B}'\bigr\} \tag{10.20}$$

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

$$U^* = U_H \cup N\bigl(\mathcal{B}'\bigr) \cup \bigl(\mathcal{B}_B \setminus \mathcal{B}'\bigr) \tag{10.21}$$

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 $m\bigl(U^*\bigr) \leq m(U)$ because of Observation 10.29, and has size

$$\bigl|U^*\bigr| \leq |U_H| + \bigl|N\bigl(\mathcal{B}'\bigr)\bigr| + \bigl|\mathcal{B}_B \setminus \mathcal{B}'\bigr| < |U_H| + \bigl|\mathcal{B}'\bigr| + \bigl|\mathcal{B}_B \setminus \mathcal{B}'\bigr| = |U| \tag{10.22}$$

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 10.28 says that U_H is the totally dominating part of U and hence that U_H is very large. 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 9.3 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, since we have chosen our parameter $C_K = 13$ for the Generalized LHC property 10.7 so generously and since the transformation in Section 9.3 works for the (non-generalized) LHC property with parameter 1, 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 9.3, 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 10.30 (Generalization of Lemma 9.30). *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 $m\bigl(Y \cup \bigcup_{i=1}^k U_i\bigr) \leq m\bigl(Y \cup \bigcup_{i=1}^k V_i\bigr)$.*

Proof. By induction over k. The base case $k = 1$ is Lemma 9.30 on page 56.

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 9.30 that

$$m\bigl(Y \cup \textstyle\bigcup_{i=1}^k U_i\bigr) = m\bigl(Y' \cup U_k\bigr) \leq m\bigl(Y' \cup V_k\bigr) = m\bigl(Y \cup \bigcup_{i=1}^{k-1} U_i \cup V_k\bigr) \ . \tag{10.23}$$

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

$$m\bigl(Y \cup \textstyle\bigcup_{i=1}^{k-1} U_i \cup V_k\bigr) = m\bigl(Y'' \cup \bigcup_{k=1}^{i-1} U_i\bigr) \leq m\bigl(Y'' \cup \bigcup_{k=1}^{i-1} V_i\bigr) = m\bigl(Y \cup \bigcup_{i=1}^{k} V_i\bigr) \tag{10.24}$$

and the lemma follows. ☐

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

Observation 10.31. *For any* $[B]\langle W\rangle \in \mathbb{S}_H$ *with* $b = \mathrm{bot}(B)$ *it holds that* $W = W \cap G^b_\vartriangle$.

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 10.15. ☐

Recalling from (10.16) that $\mathcal{W}^\vartriangle_H = \bigl\{ W \cap G^b_\vartriangle \bigm| [B]\langle W\rangle \in \mathbb{S}_H,\, b = \mathrm{bot}(B) \bigr\}$ this leads to the next, simple but crucial observation.

Observation 10.32. *The vertex set* $U_H \cup \mathcal{W}^\vartriangle_H$ *hides the vertices in* $\mathcal{B}_H$ *in the sense of Definition 9.6.*

That is, we can consider $\bigl(\mathcal{B}_H, \mathcal{W}^\vartriangle_H\bigr)$ to be (almost)[9] a standard black-white pebble configuration. This sets the stage for applying the machinery of Section 9.3.

Appealing to Lemma 9.18 on page 52, let $X \subseteq U_H \mathbin{\dot\cup} \mathcal{W}^\vartriangle_H$ be the unique, minimal tight set such that

$$\llbracket X \rrbracket = \llbracket U_H \mathbin{\dot\cup} \mathcal{W}^\vartriangle_H \rrbracket \tag{10.25}$$

and define

$$\mathcal{W}^\vartriangle_T = \mathcal{W}^\vartriangle_H \cap X \tag{10.26a}$$

$$U_T = U_H \cap X \tag{10.26b}$$

to be the vertices in $\mathcal{W}^\vartriangle_H$ and U_H that remains in X after the bottom-up pruning procedure of Lemma 9.18.

Let $\mathcal{H} = \mathcal{H}(G, X)$ be the hiding set graph of Definition 9.20 for $X = U_T \mathbin{\dot\cup} \mathcal{W}^\vartriangle_T$. Suppose that $V_1, \ldots, V_k$ are the connected components of $\mathcal{H}$, and define for $i = 1, \ldots, k$ the vertex sets

$$\mathcal{B}^i_H = \mathcal{B}_H \cap V_i \tag{10.27a}$$

$$\mathcal{W}^i_H = \mathcal{W}^\vartriangle_H \cap V_i \tag{10.27b}$$

$$U^i_H = U_H \cap V_i \tag{10.27c}$$

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

$$\mathcal{W}^i_T = \mathcal{W}^\vartriangle_T \cap V_i \tag{10.27d}$$

$$U^i_T = U_T \cap V_i \tag{10.27e}$$

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

Let us also generalize Definition 9.8 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 10.33 (Generalization of Lemma 9.36). *With the assumptions on the blob-pebbling configuration* $\mathbb{S}$ *and the vertex set* U *as in Lemma 10.26 and with notation as above, suppose that* $U^i_H \cup \mathcal{W}^i_H$ *hides* $\mathcal{B}^i_H$, *that* $\mathcal{H}\bigl(U^i_T \cup \mathcal{W}^i_T\bigr)$ *is a connected graph, and that*

$$\bigl|U^i_H\bigr| \geq 6 \cdot \bigl|\mathcal{B}^i_H \cup \mathcal{W}^i_H\bigr| \ . \tag{10.28}$$

[9] Not quite, since we might have $\mathcal{B}_H \cap \mathcal{W}^\vartriangle_H \neq \emptyset$. But at least we know that $U_H \cap \mathcal{W}^\vartriangle_H = \emptyset$ 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.

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

Proof. Let U_*^i be the set found in Lemma 9.33 on page 57, which certainly is in $[\![U_T^i \cup \mathcal{W}_T^i]\!]$, together with the prescribed extra copies of some (fixed but arbitrary) vertex that we place on level $\mathrm{maxlevel}([\![U_H^i \cup \mathcal{W}_H^i]\!]) \geq L_U$ to be on the safe side. By Lemma 9.33, U_*^i hides $\mathcal{B}_H^i$, and the size of U_*^i *counted as a multi-set with repetitions* is

$$|U_*^i| \leq |\mathcal{B}_H^i| + \lfloor |U_H^i|/3 \rfloor \leq \left(\tfrac{1}{6} + \tfrac{1}{3}\right) \cdot |U_H^i| < |U_H^i| \ . \tag{10.29}$$

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 9.36, but with the distinction 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.,

$$[\![U_T^i \dot\cup \mathcal{W}_T^i]\!] = [\![U_H^i \dot\cup \mathcal{W}_H^i]\!] \ . \tag{10.30}$$

Recall that by Definition 9.29 on page 56, 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 $m^j(U_*^i) \leq m^l(U_H^i)$. As in Lemma 9.36, we divide the proof into two cases.

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

$$\begin{aligned}
m^j(U_*^i) &= j + 2 \cdot |U_*^i\{\succeq j\}| && [\text{ by definition of } m^j(\cdot) \,] \\
&\leq j + 2 \cdot |U_*^i| && [\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V \,] \\
&\leq j + 2 \cdot \left(|\mathcal{B}_H^i| + \lfloor |U_H^i|/3 \rfloor\right) && [\text{ by Lemma 9.33 plus extra vertices }] \\
&< j + 2 \cdot |U_H^i| && [\text{ by the assumption in (10.28) }] \\
&= j + 2 \cdot |U_H^i\{\succeq j\}| && [\, U_H^i\{\succeq j\} = U_H^i \text{ since } j \leq \mathrm{minlevel}(U_H^i) \,] \\
&= m^j(U_H^i) && [\text{ by definition of } m^j(\cdot) \,]
\end{aligned}$$

and we can choose $l = j$ in Definition 9.29.

2. Consider instead $j > \mathrm{minlevel}(U_T^i \cup \mathcal{W}_T^i)$ and let $L = \mathrm{minlevel}(U_T^i \cup \mathcal{W}_T^i)$. Since the black pebbles in $\mathcal{B}_H^i$ are hidden by $U_T^i \cup \mathcal{W}_T^i$, i.e., $\mathcal{B}_H^i \subseteq [\![U_T^i \cup \mathcal{W}_T^i]\!]$ in formal notation, recollecting Definition 9.31 and Observation 9.32, part 2, we see that

$$L_{\succeq j}(\mathcal{B}_H^i) \leq L_{\succeq j}([\![U_T^i \cup \mathcal{W}_T^i]\!]) \tag{10.31}$$

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 9.34 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

$$j + L_{\succeq j}([\![U_T^i \cup \mathcal{W}_T^i]\!]) \leq L + |U_T^i \cup \mathcal{W}_T^i| \leq L + |U_H^i \cup \mathcal{W}_H^i| \ . \tag{10.32}$$

Together, (10.31) and (10.32) say that

$$j + L_{\succeq j}(\mathcal{B}_H^i) \leq L + |U_H^i \cup \mathcal{W}_H^i| \tag{10.33}$$

and using this inequality we can show that

$$\begin{aligned}
m^j(U_*^i) &= j + 2 \cdot \left|U_*^i\{\succeq j\}\right| && \big[\text{ by definition of } m^j(\cdot) \big] \\
&\leq j + L_{\succeq j}\big(\mathcal{B}_H^i\big) + \left|\mathcal{B}_H^i\right| + 2 \cdot \left\lfloor |U_H^i|/3 \right\rfloor && \big[\text{ by Lemma 9.33 + extra vertices } \big] \\
&\leq L + \left|U_H^i \cup \mathcal{W}_H^i\right| + \left|\mathcal{B}_H^i\right| + 2 \cdot \left\lfloor |U_H^i|/3 \right\rfloor && \big[\text{ using the inequality (10.33) } \big] \\
&\leq L + \tfrac{5}{3}\left|U_H^i\right| + \left|\mathcal{B}_H^i\right| + \left|\mathcal{W}_H^i\right| && \big[\, |A \cup B| \leq |A| + |B| \,\big] \\
&\leq L + \tfrac{5}{3}\left|U_H^i\right| + 2 \cdot \left|\mathcal{B}_H^i \cup \mathcal{W}_H^i\right| && \big[\, |A| + |B| \leq 2 \cdot |A \cup B| \,\big] \\
&\leq L + 2 \cdot \left|U_H^i\right| && \big[\text{ by the assumption in (10.28) } \big] \\
&= L + 2 \cdot |U_H^i\{\succeq L\}| && \big[\text{ since } L \leq \mathrm{minlevel}(U_H^i) \big] \\
&= m^L(U_H^i) && \big[\text{ by definition of } m^L(\cdot) \big]
\end{aligned}$$

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 9.29.

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 10.33. 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

$$w(V_i) = \begin{cases} \left\lceil |U_H^i|/6 \right\rceil & \text{if } \left|U_H^i\right| \geq 6 \cdot \left|\mathcal{B}_H^i \cup \mathcal{W}_H^i\right|, \\ 0 & \text{otherwise.} \end{cases} \tag{10.34}$$

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 10.34. *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

$$6 \cdot \sum_{i=1}^{k} \left|\mathcal{B}_H^i \cup \mathcal{W}_H^i\right| = 6 \cdot \left|\mathcal{B}_H \cup \mathcal{W}_H^{\triangle}\right| \leq 6 \cdot \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right| \ . \tag{10.35}$$

Since according to Corollary 10.28 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

$$\sum_{w(V_i)>0} w(V_i) \geq \sum_{w(V_i)>0} \left\lceil |U_H^i|/6 \right\rceil \geq \frac{1}{6} \cdot \left| \bigcup_{w(V_i)>0} U_H^i \right| > \left|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}_H^{\triangle}\right| \tag{10.36}$$

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 10.27 and Corollary 10.28, 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 10.33 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 10.34, 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 (10.34)). We would like to show that by judiciously performing the restructuring of Lemma 10.33 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 10.33 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 $m\big(U^*\big)$ does not exceed $m(U) = \mathrm{pot}(\mathbb{S})$ but so that $\big|U^*\big| < |U|$. But this contradicts the assumptions in Lemma 10.26. It follows that the conclusion in Lemma 10.26, 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 10.26 must have size $|U| \leq 13 \cdot \big|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}^{\triangle}_H\big|$. This in turn implies Theorem 10.19, 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 theorem. Create a weighted bipartite graph with the vertices in $\mathcal{B}_B = \big\{\mathrm{bot}(B)\big|[B]\langle W\rangle \in \mathbb{S}_B\big\}$ 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 (10.34) 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 (10.34), and each left-hand vertex has weight 1.[10] We define the neighbours of each fixed vertex $b \in \mathcal{B}_B$ to be

$$N(b) = \big\{V_i \big| w(V_i) > 0 \text{ and } \mathrm{maxlevel}\big(U^i_H\big) > \mathrm{level}(b)\big\} \ , \qquad (10.37)$$

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 10.24, 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 $w\big(N\big(\mathcal{B}'\big)\big) \leq \big|\mathcal{B}'\big|$. We must have

$$N\big(\mathcal{B}'\big) \neq \textstyle\bigcup_{i=1}^{k'} V_i \qquad (10.38)$$

since $w\big(\bigcup_{i=1}^{k'} V_i\big) > \big|\mathcal{B}_H \cup \mathcal{B}_B \cup \mathcal{W}^{\triangle}_H\big| \geq \big|\mathcal{B}_B\big|$ by Proposition 10.34. For all $\mathcal{B}'' \subseteq \mathcal{B}_B \setminus \mathcal{B}'$ it holds that

$$w\big(N\big(\mathcal{B}''\big) \setminus N\big(\mathcal{B}'\big)\big) \geq \big|\mathcal{B}''\big| \qquad (10.39)$$

since otherwise $\mathcal{B}'$ would not be of largest size as assumed above. The inequality (10.39) 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

[10] 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 (10.37) always containing either all or none of these vertices.

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 in $\mathcal{H}$ are $V_1, \ldots, V_m$ and the rest of the components are $V_{m+1}, \ldots, V_k$ and so that $U_H^1, \ldots, U_H^m$ and $U_H^{m+1}, \ldots, U_H^k$ are the corresponding subsets of the hiding set U_H. Then pick good local blockers $U_*^i \subseteq V_i$ as in Lemma 10.33 for all components $V_1, \ldots, V_m$. Now the following holds:

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

2. All subconfigurations in
$$\mathbb{S}_B^1 = \{[B]\langle W\rangle \in \mathbb{S}_B \,|\, \mathrm{bot}(B) \in \mathcal{B}'\} \tag{10.40}$$
are blocked by $U_B \cup N(\mathcal{B}') = U_B \cup \bigcup_{i=m+1}^{k} U_H^i$, as we have not moved any elements in U above $\mathcal{B}'$.

3. With notation as in Lemma 10.30, 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 10.33. 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 10.30 are satisfied and we conclude that
$$\begin{aligned} m\big(U_B \cup \textstyle\bigcup_{i=1}^{m} U_*^i \cup \bigcup_{i=m+1}^{k} U_H^i\big) &= m\big(Y \cup \textstyle\bigcup_{i=1}^{m} U_*^i\big) \\ &\leq m\big(Y \cup \textstyle\bigcup_{i=1}^{m} U_H^i\big) \\ &= m\big(U_B \cup \textstyle\bigcup_{i=1}^{m} U_H^i \cup \bigcup_{i=m+1}^{k} U_H^i\big) \\ &= m(U) \enspace , \end{aligned} \tag{10.41}$$
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
$$\big|U_B \cup \textstyle\bigcup_{i=1}^{m} U_*^i \cup \bigcup_{i=m+1}^{k} U_H^i\big| < |U| \enspace , \tag{10.42}$$
where again the multi-set is *counted with repetitions*.

4. It remains to take care of the potentially unblocked subconfigurations in
$$\mathbb{S}_B^2 = \{[B]\langle W\rangle \in \mathbb{S}_B \,|\, \mathrm{bot}(B) \in \mathcal{B}_B \setminus \mathcal{B}'\} \enspace . \tag{10.43}$$
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
$$w(V_i) = \lceil |U_H^i|/6 \rceil \leq \lfloor |U_H^i|/3 \rfloor \tag{10.44}$$
vertices from $\mathcal{B}_B \setminus \mathcal{B}'$ (where we used that $|U_H^i| \geq 6$ 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 $\mathrm{maxlevel}(U_H^i) > \mathrm{level}(b)$. By Observation 10.29, lowering these spare vertices from $\mathrm{maxlevel}(U_H^i)$ 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 $m(U^*) \leq m(U)$ but $|U^*| < |U|$. This is a contradiction since U was chosen to be of minimal size, and thus Lemma 10.26 must hold. But then Theorem 10.19 follows immediately as well, as was noted above.

10.5 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.

Theorem 10.35 (rephrasing of Theorem 1.1). *Suppose that G_h is a layered blob-pebblable 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 $Sp(Peb^d_{G_h} \vdash 0) = \Theta(h)$.*

Proof. The $\mathrm{O}(h)$ upper bound on clause space follows from the bound $\mathit{Peb}(G_h) \leq h + \mathrm{O}(1)$ on the black pebbling price in Lemma 9.2 on page 45 combined with the bound $Sp(Peb^d_G \vdash 0) \leq \mathit{Peb}(G) + \mathrm{O}(1)$ from Proposition 4.15 on page 17.

For the lower bound, we instead consider the pebbling formula $*Peb^d_{G_h}$ without target axioms $\overline{x(z)}_1, \ldots, \overline{x(z)}_d$ and use that by Lemma 7.1 on page 30 it holds that $Sp\bigl(Peb^d_{G_h} \vdash 0\bigr) = Sp\bigl(*Peb^d_{G_h} \vdash \bigvee_{i=1}^{d} x(z)_i\bigr)$. 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 7.3 on page 31 such that $\mathit{cost}(\mathcal{P}_\pi) \leq \max_{\mathbb{C} \in \pi}\bigl\{\mathit{cost}(\mathbb{S}(\mathbb{C}))\bigr\} + \mathrm{O}(1)$. On the one hand, Theorem 8.5 on page 41 says that $\mathit{cost}(\mathbb{S}(\mathbb{C})) \leq |\mathbb{C}|$ provided that $d > 1$, so in particular it must hold that $\mathit{cost}(\mathcal{P}_\pi) \leq Sp(\pi) + \mathrm{O}(1)$. On the other hand, $\mathit{cost}(\mathcal{P}_\pi) \geq \mathit{Blob\text{-}Peb}(G_h)$ by definition, and by Theorems 10.8 and 10.19 it holds that $\mathit{Blob\text{-}Peb}(G_h) = \Omega(h)$. Thus $Sp(\pi) = \Omega(h)$, and the theorem follows. □

Plugging in pyramid graphs Π_h in Theorem 10.35, we get k-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 10.36. *Let G be any layered spreading graph and suppose that Peb^d_G has formula size and number of clauses $\Theta(n)$. Then $Sp\bigl(Peb^d_G \vdash 0\bigr) = \mathrm{O}(\sqrt{n})$.*

Proof. Suppose that G has height h. Then $Sp\bigl(Peb^d_G \vdash 0\bigr) = \mathrm{O}(h)$ 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 $|V(G)| = \Omega\bigl(h^2\bigr)$, 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 9.23, and setting $j = h$ in the spreading inequality (9.10) we get that $\left|V'_L\right| \geq 1 + h - L$. Hence $|V(G)| \geq \sum_{L=0}^{h} |V'_L| = \Omega\bigl(h^2\bigr)$. □

The proof of Theorem 10.36 can also be extended to cover the original definition in [37] of spreading graphs that are not necessarily layered, but we omit the details.

11 Conclusion and Open Problems

We have proven an asymptotically tight bound on the refutation clause space in resolution of pebbling contradictions over pyramid graphs. This yields the currently best known separation 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 corresponding formulas. Instead, a corollary of our result is an exponential improvement of the separation of width and space in [42]. This is a first step towards answering the question of the relationship between length and space posed in, for instance, [11, 29, 57].

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 asymptotical lower bound for the refutation clause space $Sp\left(Peb_G^d \vdash 0\right)$ of pebbling contradictions Peb_G^d provided that $d \geq 2$. Plugging in pyramid graphs we get an $\Omega(\sqrt{n})$ 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 [31] of size $\mathrm{O}(n)$ that have maximal black-white pebbling price *BW-Peb*$(G_n) = \Omega(n/\log n)$ in terms of size? If it could be proven for pebbling contradictions over such graphs that pebbling price bounds clause space from below, this would immediately imply that there are k-CNF formulas refutable in small length that can be maximally complex with respect to clause space.

Open Problem 1. *Is there a family of unsatisfiable k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ such that $L(F_n \vdash 0) = \mathrm{O}(n)$ and $W(F_n \vdash 0) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Omega(n/\log n)$?*

We are currently working on this problem, but note that these DAGs in [31] seem to have much more challenging structural properties that makes it hard to lift the lower bound argument from standard black-white pebblings to blob-pebblings.

A second question, more related to Theorem 1.3 and the other trade-off results presented in Section 5, is as follows. We know from [15] (see Theorem 4.2) 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 [15] 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 valuable 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 [15] just an artifact of the proof?

Open Problem 2. *If F is a k-CNF formula over n variables refutable in length L, is it true that there is always a refutation π of F in width $W(\pi) = \mathrm{O}\left(\sqrt{n \log L}\right)$ with length no more than, say, $L(\pi) = \mathrm{O}(L)$ or at most $\mathrm{poly}(L)$?*

A similar trade-off question can be posed for clause space. Given a refutation in small space, we can prove using [5] (see Theorem 4.5) that there must exist a refutation in short length. But again, the short refutation resulting from the proof is not the same as that with which we started. For concreteness, let us fix the space to be constant. If a polynomial-size k-CNF formula has a refutation in constant clause space, we know that it must be refutable in polynomial length. But can we get a refutation in both short length and small space simultaneously?

Open Problem 3. *Suppose that $\{F_n\}_{n=1}^{\infty}$ is a family of polynomial-size k-CNF formulas with refutation clause space $Sp(F_n \vdash 0) = \mathrm{O}(1)$. Does this imply that there are refutations $\pi_n : F_n \vdash 0$ simultaneously in length $L(\pi_n) = \mathrm{poly}(n)$ and clause space $Sp(\pi_n) = \mathrm{O}(1)$?*

Or can it be that restricting the clause space, we sometimes have to end up with really long refutations? We would like to know what holds in this case, and how it relates to the trade-off results for variable space in [33].

Finally, we note that all bounds on clause space proven so far is in the regime where the clause space $Sp(\pi)$ is less than the number of clauses $|F|$ in F. This is quite natural, since the size of the formula can be shown to be an upper bound on the minimal clause space needed [28].

Such lower bounds on space might not seem too relevant to clause learning algorithms, since the size of the cache in practical applications usually will be very much larger than the size of the formula. For this reason, it seems to be a highly interesting problem to determine what can be said if we allow extra clause space. Assume that we have a CNF formula F of size roughly n refutable in length $L(F \vdash 0) = L$ for L suitably large (say, $L = \mathrm{poly}(n)$ or $L = n^{\log n}$ or so). Suppose that we allow clause space more than the minimum $n + \mathrm{O}(1)$, but less than the trivial upper bound $L/\log L$. Can we then find a resolution refutation using at most that much space and achieving at most a polynomial increase in length compared to the minimum?

Open Problem 4 ([12]). *Let F be any CNF formula with $|F| = n$ clauses (or $|Vars(F)| = n$ variables). Suppose that $L(F \vdash 0) = L$. Does this imply that there is a resolution refutation $\pi : F \vdash 0$ in clause space $Sp(\pi) = \mathrm{O}(n)$ and length $L(\pi) = \mathrm{poly}(L)$?*

If so, this could be interpreted as saying that a smart enough clause learning algorithm can potentially find any short resolution refutation in reasonable space (and for formulas that cannot be refuted in short length we cannot hope to find refutations efficiently anyway).

We conclude with a couple of comments on clause space versus clause learning.

Firstly, we note that it is unclear whether one should expect any fast progress on Open Problem 4, at least if if our experience from the case where $Sp(\pi) \leq |F|$ is anything to go by. Proving lower bounds on space in this "low-end regime" for formulas easy with respect to length has been (and still is) very challenging. However, it certainly cannot be excluded that problems in the range $Sp(\pi) > |F|$ might be approached with different and more successful techniques.

Secondly, we would like to raise the question of whether, in spite of what was just said before Open Problem 4, lower bounds on clause space can nevertheless give indications as to which formulas might be hard for clause learning algorithms and why. Suppose that we know for some CNF formula F that $Sp(F \vdash 0)$ 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 sure that the fairly large critical set of clauses needed to find a refutation will be among the "active" clauses.

Intriguingly enough, pebbling contradictions over pyramids might in fact be an example of this. 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 [52], it was shown that state-of-the-art clause learning algorithms can have serious problems with even moderately large pebbling contradictions.[11] Although we are certainly not arguing that this is the whole story—it was also shown in [52] that the branching order is a critical factor, and that given some extra structural information the algorithm can achieve an exponential speed-up—we wonder whether the high lower bound on clause space can nevertheless be part of the explanation. It should be pointed out that pebbling contradictions are the only formulas we know of that are really easy with respect to length and width but hard for clause space. And if there is empirical data showing that for these very formulas clause learning algorithms can have great difficulties finding refutations, it might be worth investigating whether this is just a coincidence or a sign of some deeper connection.

Acknowledgements

We are grateful 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

[11] The "grid pebbling formulas" in [52] are exactly our pebbling contradictions of degree $d = 2$ over pyramid graphs.

blob-pebble game. Also, we would like to thank Paul Beame, Maria Klawe, Philipp Hertel, and Toniann Pitassi 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 11 in particular.

References

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

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

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

[4] Noga Alon and Michael Capalbo. Smaller explicit superconcentrators. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, pages 340–346, 2003.

[5] Albert Atserias and Victor Dalmau. A combinatorical characterization of resolution width. In *Proceedings of the 18th IEEE Annual Conference on Computational Complexity (CCC '03)*, pages 239–247, July 2003. Journal version to appear in *Journal of Computer and System Sciences*.

[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, June 2001.

[7] Paul Beame. Proof complexity. In Steven Rudich and Avi Wigderson, editors, *Computational Complexity Theory*, volume 10 of *IAS/Park City Mathematics Series*, pages 199–246. American Mathematical Society, 2004.

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

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

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

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

[12] Eli Ben-Sasson. Personal communication, 2007.

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

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

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

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

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

[18] Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, December 2001.

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

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

[21] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC '71)*, pages 151–158, 1971.

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

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

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

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

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

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

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

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

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

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

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

[33] Philipp Hertel and Toniann Pitassi. Exponential time/space speedups for resolution and the PSPACE-completeness of black-white pebbling. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*, pages 137–149, October 2007.

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

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

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

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

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

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

[40] Friedhelm Meyer auf der Heide. A comparison of two variations of a pebble game on graphs. *Theoretical Computer Science*, 13(3):315–322, 1981.

[41] Jakob Nordström. *Narrow Proofs May Be Spacious: Separating Space and Width in Resolution*. Technical Report TR05-066, Revision 02, Electronic Colloquium on Computational Complexity (ECCC), November 2005.

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

[43] Jakob Nordström. *A Simplified Way of Proving Trade-off Results for Resolution*. Technical Report TR07-114, Electronic Colloquium on Computational Complexity (ECCC), September 2007.

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

[45] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[46] Christos H. Papadimitriou and David Wolfe. The complexity of facets resolved. *Journal of Computer and System Sciences*, 37(1):2–13, 1988.

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

[48] 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.

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

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

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

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

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

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

[55] Gunnar Stålmarck. Short resolution proofs for a sequence of tricky formulas. *Acta Informatica*, 33(3):277–280, May 1996.

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

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

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

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

ECCC ISSN 1433-8092

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