New Wine into Old Wineskins: A Survey of Some Pebbling Classics with Supplemental Results

Jakob Nordström
KTH Royal Institute of Technology*
100 44 Stockholm, Sweden
jakobn@kth.se

November 19, 2010

Abstract

Pebble games were extensively studied in the 1970s and 1980s in a number of different contexts. The last decade has seen renewed interest in pebbling in the field of proof complexity. This is a survey of some classical theorems in pebbling, as well as a couple of new ones, with a focus on results that have proven relevant in proof complexity applications.

THIS IS A MANUSCRIPT IN PREPARATION. See Section 1.4 on page 6 for a report on the status of the different sections of the paper. Questions, corrections, clarifications or any other comments are most welcome!

Contents

*Part of this work performed while at the Massachusetts Institute of Technology supported by grants from the Royal Swedish Academy of Sciences, the Ericsson Research Foundation, the Sweden-America Foundation, the Foundation Olle Engkvist Byggmästare, the Sven and Dagmar Salén Foundation, and the Foundation Blanceflor Boncompagni-Ludovisi, née Bildt.

1 Introduction

Pebbling is a tool for studying the relationship between time and space by means of a game played on directed acyclic graphs. The pebble game models computations where the execution is independent of the input and can be performed by straight-line programs. Each such program is encoded as a graph, and a pebble on a vertex in the graph indicates that the corresponding value is currently kept in memory. The goal is to pebble the output vertex of the graph with number of pebbles (amount of memory) and steps (amount of time) that are minimal.

Pebble games were originally devised for studying programming languages and compiler construction, but have later found a broad range of applications in computational complexity theory. The pebble game model seems to have appeared for the first time (implicitly) in [PH70], where it was used to study flowcharts and recursive schemata, and it was later employed to model register allocation [Set75], and analyze the relative power of time and space as Turing-machine resources [Coo74, HPV77]. Moreover, pebbling has been used to derive time-space trade-offs for algorithmic concepts such as linear recursion [Cha73, SS83], fast Fourier transform [SS77, Tom78], matrix multiplication [Tom78], and integer multiplication [SS79].

An excellent survey of pebbling up to ca 1980 is [Pip80], and the introduction in [LT82] also contains much information. Another in-depth treatment of some pebbling-related questions can be found in chapter 10 of [Sav98]. This introductory section is heavily indebted to these three sources.

In the last decade, there has been a renewed interest in pebbling in the context of proof complexity. A list of papers in proof complexity (without any claim of being exhaustive) that have used pebbling in one way or another is [AJPU07, BEGJ00, BIPS10, BCIP02, BP07, Ben09, BIW04, BN08, BN11, BW01, EGM04, ET01, ET03, HP10, HU07, Kul99, Nor09a, Nor10a, NH08b, SBK04]. Out of space considerations, proof complexity will be discussed fairly briefly and selectively in this survey. We refer to, for instance, [Bea04, BP98, CK02, Seg07, Urq95] for more information about this field of research.

1.1 Pebbling in a Nutshell

The *pebbling price* of a directed acyclic graph G in the traditional black pebble game captures the memory space, or number of registers, required to perform the deterministic computation described by G. We will mainly be interested in the the more general *black-white pebble game* modelling nondeterministic computation, which was introduced in [CS76] and has been studied in [GT78, Kla85, LT80, LT82, Mey81, KS91] and other papers.

Definition 1.1 (Pebble game). Let G be a directed acyclic graph (DAG) with a unique sink vertex z. The *black-white pebble game* on G is the following one-player game. At any time t, we have a configuration $\mathbb{P}_t = (B_t, W_t)$ of black pebbles B_t and white pebbles W_t on the vertices of G, at most one pebble per vertex. 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 a source vertex.

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 *(complete) black-white pebbling of* G, also called a *pebbling strategy for* G, is a sequence of pebble configurations $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = (\emptyset, \emptyset)$, $\mathbb{P}_\tau = (\{z\}, \emptyset)$, and for all $t \in [\tau]$, $\mathbb{P}_t$ follows from $\mathbb{P}_{t-1}$ by one of the rules above. The *time* of a pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is simply $\mathit{time}(\mathcal{P}) = \tau$ and the *space* is $\mathit{space}(\mathcal{P}) = \max_{0 \leq t \leq \tau}\{|B_t \cup W_t|\}$. The *black-white pebbling price* (also known as the *pebbling measure* or *pebbling number*) of G, denoted $\mathit{BW\text{-}Peb}(G)$, is the minimum space of any complete pebbling of G.

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

See Figure 1 for an example of a complete black-white pebbling.

Obviously, any DAG G over n vertices can be pebbled in linear time simply by placing black pebbles on all vertices in topological order, and this is also a lower bound. Thus, studying the time measure in isolation is not that exciting. More interesting is to investigate the space needed for pebbling graphs, and how time and space are related if one wants to optimize both measures simultaneously in a pebbling.

We already noted that any graph G with n vertices can be completely black-pebbled with n pebbles in time $2n$. Another easy observation is that if G can be pebbled with s pebbles at all, then such a pebbling cannot take more time than

$$2 \sum_{r=0}^{s} 2^r \binom{n}{r} \leq 2^{2n+1} , \tag{1.1}$$

since each possible distinct black-white pebble configuration need only appear and disappear once during a pebbling (otherwise the pebbling can be shortened by omitting the moves between two repetitions of a pebble configuration). Hence, the range of interest for time is between $\mathrm{O}(n)$ and $\exp(\mathrm{O}(n))$, and for space the interesting range is between $\mathrm{O}(1)$ and $\mathrm{O}(n)$. The focus of much pebbling research has been to investigate the relative power of time and space within these intervals.

Perhaps the first result that really drew attention to pebbling was that of Hopcroft et al. [HPV77], who proved that any DAG with n vertices of bounded indegree can be black-pebbled with $\mathrm{O}(n/\log n)$ pebbles. They used this result to establish that space is strictly stronger than time as a computational resource for Turing machines. Paul et al. [PTC77] showed that this upper bound is tight by exhibiting families of DAGs requiring $\Omega(n/\log n)$ black pebbles, and Gilbert and Tarjan [GT78] extended this lower bound to the black-white pebble game, albeit with different constants.

The reduction of space from $\mathrm{O}(n)$ to $\mathrm{O}(n/\log n)$ in [HPV77] comes at the price of an exponential increase in the pebbling time. It is natural to ask whether this is necessary, and how much space can be saved in general while still keeping the pebbling time polynomial in n (adopting the general paradigm in theoretical computer science that polynomial time corresponds to feasible computations). Pippenger [Pip78], Tompa [Tom78], and Reischuk [Rei80] presented graphs for which the pebbling time increases to superlinear (but still polynomial) as the space is decreased from $\mathrm{O}(n)$ to $\mathrm{O}(n/\log n)$. Reischuk [Rei80] also showed that any DAG can be pebbled in space $\mathrm{O}(n/\log\log\log n)$ and polynomial time. In their seminal paper [LT82], Lengauer and Tarjan resolved this question by establishing that there are constants $K_1 \leq K_2$ such that the pebbling space can always be brought down to $K_2 \cdot n/\log\log n$ while still keeping the time polynomial, but for space $K_1 \cdot n/\log\log n$ there are graphs that require superpolynomial time. More generally, they showed that for pebbling space in the range from $s = \Omega(n/\log n)$ to $s = \mathrm{O}(n)$, pebbling time $s \exp\exp(\mathrm{O}(n/s))$ is sufficient for any graph, but that fixing n and s one can find graphs for which pebbling

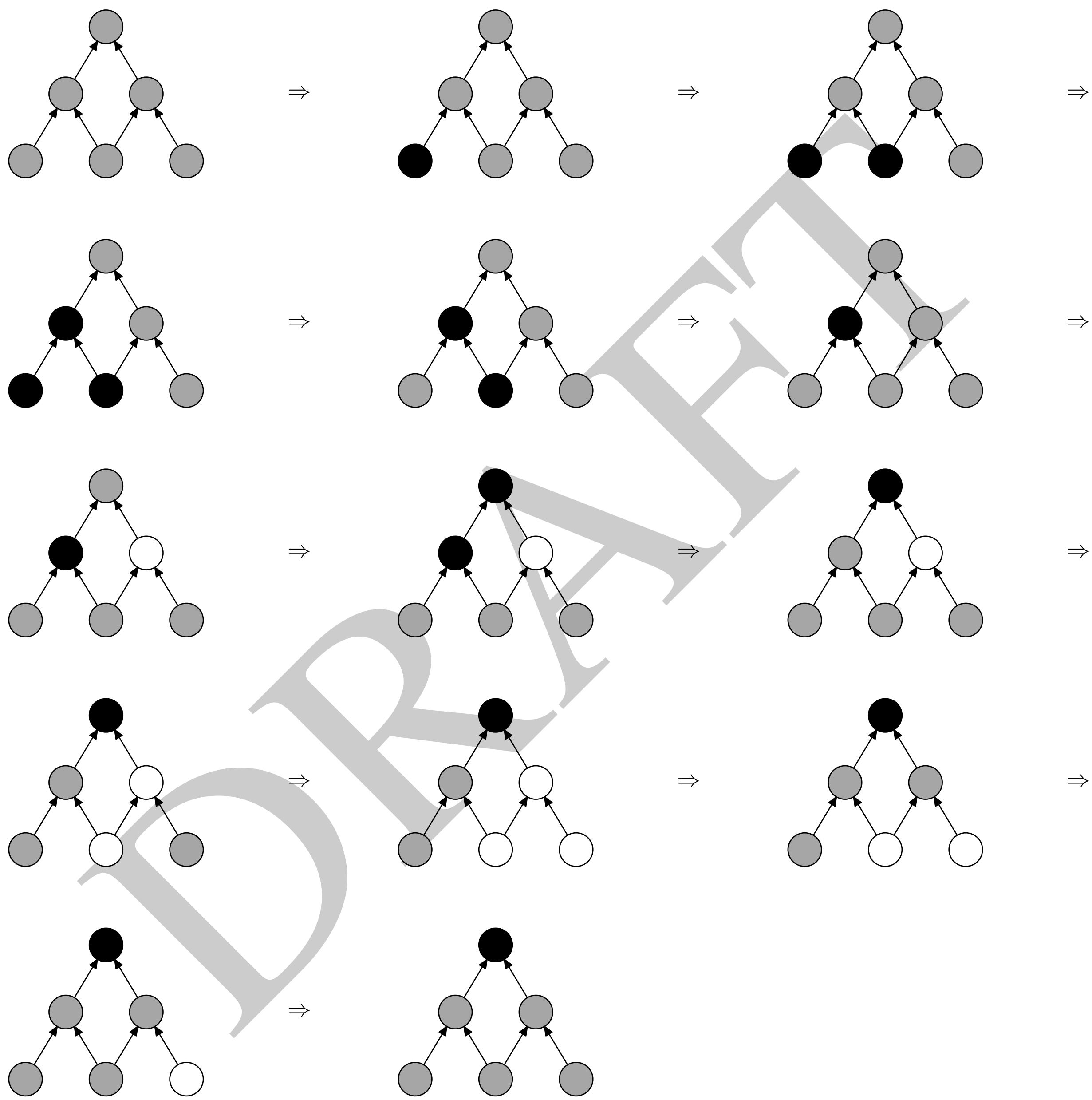


Figure 1: Complete black-white pebbling of pyramid of height 2.

time $s \exp\exp(\Omega(n/s))$ is necessary. In particular, their construction yields explicit graphs for which any pebbling in space $\mathrm{O}(n/\log n)$ must take exponential time (by which we mean $\exp(n^{\epsilon})$ for some $\epsilon > 0$). The results in [LT82] hold for both black and black-white pebbling.

Carlson and Savage [CS80, CS82] focused on the other endpoint of the space interval and considered how reductions in space affect the pebbling time for graphs pebblable in very small space. They gave an explicit and elegant construction of graph families with arbitrarily small non-constant pebbling price for which this space can only be achieved by pebblings in superpolynomial (but subexponential) time. They also proved that there are graph families exhibiting very robust trade-offs in the sense that the requirement of time growing faster than any polynomial is sustained over a very broad range of space. More precisely, they exhibit graphs for which the pebbling price is only logarithmic in the number of vertices n but for which the space has to be increased to essentially $\sqrt{n}$ in order to pebble the graphs in polynomial time. Carlson and Savage dealt only with black pebbling, but their results were recently extended to black-white pebbling in [Nor10a]. Lengauer and Tarjan studied this low end of the space interval as well in [LT82], proving polynomial trade-offs for constant space and robust superpolynomial (but subexponential) trade-offs for space between $\Omega(\log^2 n)$ and $\mathrm{O}(n/\log n)$. All of these trade-offs in [CS80, CS82, LT82] are essentially the best one can hope for in their respective space intervals, since exponential trade-offs are ruled out by the simple counting argument used to obtain the upper bound in Equation (1.1).

Yet another direction was taken in the papers [Lin78, PT78, EBL79, GLT80], which show explosive pebbling time increases from linear to exponential when the pebbling space drops by a (small) constant amount. This happens for space in the range $\Theta(n^k) = \mathrm{o}(n/\log n)$ for different k, $1/4 \leq k \leq 1/2$, in the different papers, i.e., somewhere in the middle of the space interval between constant and linear.

Wrapping up this short and somewhat selective introduction to the pebbling literature, we also want to mention that a number of papers have investigated the pebbling price of concrete, simple families of graphs, for instance, [Coo74, CS76, Kla85, LT80]. In general, though, determining the black pebbling price of a graph is PSPACE-complete [GLT80], and in the recent paper [HP10], this result was extended to black-white pebbling as well, albeit with the technical restriction that the graphs must have unbounded fan-in. Finally, we note that although for almost all graphs studied in this survey the black and black-white pebbling prices coincide asymptotically, this is not the case in general. Meyer auf der Heide [Mey81] has shown that the difference in black and black-white pebbling price can be at most quadratic, and Kalyanasundaram and Schnitger [KS91] have proven an asymptotically matching lower bound.

1.2 Outline of this Survey

This survey is organized as follows. In Section 2 we fix the terminology and notation for graphs and then give some formal definitions of different flavours of the pebble game that occur in the literature, discussing how these variants of pebbling are related to each other. We continue with the preliminaries in Section 3, where we focus on some technical pebbling tools that will be used extensively in the proofs. We then turn to examining some graph families that have been used as building blocks in many pebbling results. In Section 4 we discuss layered graphs (in particular, trees and pyramids) drawing on material in [Coo74, CS76, Kla85, LT80], and in Section 5 we study so-called superconcentrators, following the exposition in [LT82].

With the material from Sections 3, 4, and 5 in hand, we can state and prove the main pebbling results in this survey. Section 6 presents the two general upper bounds on pebbling space and time-space trade-offs from [HPV77, LT82], using the simplified proofs of [Lou80]. In Section 7, we show a matching lower bound on space from [GT78, PTC77]. We then prove a series of trade-off results. Section 8 presents polynomial time-space trade-offs for constant pebbling space. In Section 9, we establish that as soon as one goes above constant space, there are superpolynomial trade-offs. In Section 10, we show that there are very robust trade-offs in the sense that the graphs are pebblable in very small space, but to avoid the superpolynomial time penalty in the trade-off one has to go almost all the way up to linear space. Section 11 contains an exposition of exponential time-space trade-offs. These four sections build on material from [CS80, CS82, LT82, Nor10a]. We point out that for some of the applications in proof complexity, we are interested in pebbling trade-offs of an especially strong kind, where the lower bounds hold for black-white pebbling while there are matching upper bounds for black pebbling. Most of the trade-off results presented in the current paper are of this type. We also remark that all trade-off results are for explicitly constructible graphs.

Concluding the survey proper about pebbling, in Section 12, we discuss what is known about the relative strength of black-white and black-only pebbling. This paper cannot hope to be an exhaustive survey of all pebbling results, however, and in Section 13 we mention some important results that we do *not* cover in detail.

In Section 14, we describe some connections to proof complexity, focusing on explaining how pebble games can be used to derive results for the resolution proof system.

As we go along in the paper, we mention some pebbling-related problems that have remained open. These problems, as well as some related open problems in proof complexity, are all collected for easy reference in Section 15, which concludes the paper.

1.3 Rationale for and Contribution of This Survey

This survey attempts to present some of the main results in the pebbling literature in as clear a fashion as possible and in a unified way, avoiding subtleties introduced by slightly different definitions of pebbling in different papers. We have striven to simplify the expositions of the proofs when possible (and fix some minor bugs, hopefully without introducing too many new ones), and also provide ample illustrations of the various graphs used in the constructions.

Comparing this survey to [Pip80] and [Sav98], it should be said that Chapter 10 in Savage [Sav98] only deals with the black pebble game and in this sense has a slightly more narrow focus, but also contains a wealth of material not covered here, whereas Pippenger [Pip80] has a wider scope than this survey, but for natural reasons contains no material about developments since 1980. Also, [Pip80] is fairly concise and in particular provide very little details about proof techniques, whereas the current paper contains proofs or at least detailed proof sketches for all results. One reason why this can be of interest is that in some proof complexity papers (notably, [Nor09a, NH08b, BN11]) the pebbling results cannot be used in a "black-box fashion", but instead one has too dig fairly deep into the technical constructions.

Finally, we hope that the overview of how pebble games have been used in proof complexity over the last decade can serve as a useful reference, and perhaps stimulate more research into the connections between proof complexity and pebbling.

1.4 Status of This Write-up

This is a manuscript in preparation. The status of the different sections of the paper as of November 19, 2010, is as follows:

- Section 1: Finished (except that Section 1.4 should be erased at some point).
- Sections 2, 3, and 4: Finished.
- Section 5: Essentially finished. First (nearly-)final version written, but might contain some rough edges.
- Sections 6, 7, 8, and 9: Finished.
- Section 10: The main formal statements are in place but almost all details are missing.
- Section 11: Some tentative formal statements are in place, but need to be checked. Basically nothing is written here.
- Sections 12, 13, and 14: Finished.
- Section 15: Raw material in place, but writing and editing remains to be done.

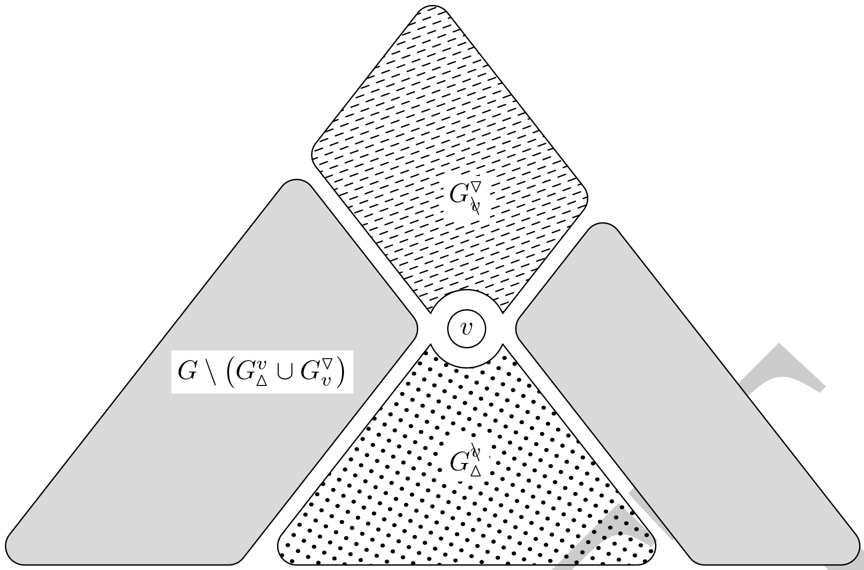


Figure 2: Notation for sets of vertices in DAG G with respect to vertex v.

2 Basic Definitions and Some Easy Facts

We first fix notation and terminology for graphs, and then provide some pebbling definitions and state some easy facts about them. Let us note right away that all logarithms in this paper are base 2 unless otherwise specified, and that $[n]$ denotes the set of integers $\{1, 2, \ldots, n\}$.

2.1 Some Graph Notation and Terminology

We write G to denote a graph with vertices $V(G)$ and edges $E(G)$. All graphs in this paper are directed unless otherwise stated, and (u, v) denotes a directed edge from u to v.

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 transitive closures of $succ(\cdot)$ and $pred(\cdot)$, we let $G^{\triangledown}_{v}$ denote all vertices reachable from v (vertices "above" v) and $G^{v}_{\vartriangle}$ denote all vertices from which v is reachable (vertices "below" v). We write $G^{\not v}_{\vartriangle}$ and $G^{\triangledown}_{\not v}$ to denote the corresponding sets with the vertex v itself removed. See Figure 2 for an illustration of this notation. If $u, w \in pred(v)$, we say that u and w are *siblings*. If $u \notin G^{w}_{\vartriangle}$ and $w \notin G^{u}_{\vartriangle}$, we say that u and w are *non-comparable* vertices. Otherwise they are *comparable*.

For brevity, we will overload the notation $G^{v}_{\vartriangle}$ and $G^{\triangledown}_{v}$ to refer not only to subsets of vertices of G but also to the subgraphs of G induced on these subsets (i.e., containing all edges in $E(G)$ between vertices in the subsets). Moreover, when no misunderstanding can occur we will sometimes overload the notation for the graph G itself and its vertices, and write only G when we mean $V(G)$.

We say that vertices of G with indegree 0 are *sources* and that vertices with outdegree 0 are *sinks*. In the literature, sources are also referred to as *inputs* and sinks as *targets* or *outputs*, but we will try to stick to our terminology throughout this survey. In the notation introduced above, a source vertex s in G is a vertex with $pred(s) = \emptyset$, and for a sink z we have $succ(z) = \emptyset$. We will write $S(G)$ to denote the source vertices of G and $Z(G)$ to denote the sink vertices. For brevity, we will sometimes refer to a DAG with a unique sink as a *single-sink DAG*.

Some more notational conventions are that the parameter ℓ denotes the maximal indegree of a DAG, and that when not stated otherwise, n will denote the size, i.e., the number of vertices, of a DAG (or, in some cases where it is more convenient, the size to within a small constant factor). We write $Q : v \rightsquigarrow w$ to denote a path Q starting in the vertex v and ending in the vertex w. A *source path* is a path that starts at some source vertex of G. A *path via* u is a path Q such that $u \in Q$. We will also say that Q *visits* u.

For "simple" graphs that we will use as building blocks in our constructions, such as pyramids or superconcentrators (to be defined later), we will use the convention that the subindex j in G_j gives an indication of the size of the graph in question. For instance, Π_h will denote a pyramid of height h. In more complicated constructions, the notation $G^{(1)}, G^{(2)}$ et cetera signifies that we pick identical copies of a graph G indexed by the superindices.

2.2 Definitions of Pebble Games

We already described pebbling in Definition 1.1, which is adapted from [CS76] although it uses the established pebbling terminology introduced by [HPV77]. The flavour of the pebble game formalized there is the version that we are interested in for our applications in proof complexity, but for the purposes of stating and proving the results covered in this survey we need a slightly more general definition.

Definition 2.1 (General pebbling definition). Suppose that G is a DAG with sources S and sinks Z (one or many). A *black-white pebbling* from (B_0, W_0) to (B_τ, W_τ) in G is a sequence of pebble configurations $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ such that $\mathbb{P}_0 = (B_0, W_0)$, $\mathbb{P}_\tau = (B_\tau, W_\tau)$, and for all $t \in [\tau]$, $\mathbb{P}_t$ follows from $\mathbb{P}_{t-1}$ by one of the rules in Definition 1.1. The space of a pebble configuration $\mathbb{P} = (B, W)$ is $\mathit{space}(\mathbb{P}) = |B \cup W|$ and the space of the pebbling $\mathcal{P}$ is $\mathit{space}(\mathcal{P}) = \max_{t \in [\tau]}\{\mathit{space}(\mathbb{P}_t)\}$.

We say that a pebbling $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ is *conditional* if $\mathbb{P}_0 \neq (\emptyset, \emptyset)$ and *unconditional* otherwise. Note that complete pebblings, or pebbling strategies, as defined in Definition 1.1 are always unconditional.

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

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

For conciseness, sometimes it will be convenient to refer to a pebbling (conditional or unconditional) that touches all sinks as a *sink pebbling*. Thus, a complete pebbling is a unconditional sink pebbling.

We note that many pebbling papers focus exclusively on either visiting or persistent pebblings, and there seems to be no standard terminology for distinguishing between them. Loui [Lou79] refers to visiting pebblings of a graph G with sink z as pebblings *promising* z and to persistent pebblings as pebblings *ensuring* z, but this usage appears not to have caught on. Hopefully the new terminology and notation proposed in Definition 2.1 will be intuitive and easy to understand.

We will use the parameter s to denote pebbling space (although $s_1, s_2, \ldots$ will sometimes denote source vertices of DAGs). 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 $[t_1, t_2]$".[1]

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

[1] We remark that some papers measure time as the number of pebble placements, ignoring removals. This convention is natural in the sense that the most straightforward approach when proving pebbling lower bounds is to count the number of pebble placements. The reason that we prefer to count both placements and removals is that we find it very convenient to be able to discuss what happens "at time t" in the pebbling regardless of what kind of move this is. Having every placement and removal correspond to a time step makes writing the proofs easier.

easy to see that $\mathit{Peb}^{\bullet}(G) = \mathit{Peb}^{\emptyset}(G)$. In asymptotic statements when the above differences do not matter, we will just write $\mathit{BW\text{-}Peb}(G)$ and $\mathit{Peb}(G)$.

The only pebblings we are really interested in are complete pebblings of G. However, when proving lower bounds on pebbling price it will sometimes be convenient to be able to reason in terms of partial pebbling move sequences, i.e., conditional pebblings. One can think of conditional pebblings as pebblings that receive the start configuration (B_1, W_1) "as a gift", and are also allowed to leave (B_2, W_2) without "cleaning up" when they finish. It is clear that we can assume that $(B_1, W_1) = (B_1, \emptyset)$ and $(B_2, W_2) = (\emptyset, W_2)$ since we can freely place white pebbles on G and freely remove black pebbles. The way the gift can help us is that we get black pebbles at the beginning for free, and are allowed to leave white pebbles without having to do the hard pebbling work of removing them.

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

Observation 2.2. *Suppose that G is a DAG and that $\mathcal{P}$ is any complete pebbling of G. Let $U \subseteq V(G)$ be any subset of vertices of G and let $H = H(G, U)$ denote the induced subgraph with vertices $V(H) = U$ and edges $E(H) = \{(u, v) \in E(G) \mid u, v \in U\}$. Then the pebbling $\mathcal{P}$ restricted to the vertices in U is a complete visiting pebbling strategy for H.*

Proof. It is easy to verify that if we only perform those pebbling moves in $\mathcal{P}$ that pertain to vertices in U, then these moves constitute a legal pebbling on H in the sense of Definition 1.1. Moreover, any complete pebbling of G must pebble all vertices in G, so $\mathcal{P}$ restricted to U will pebble all vertices in H including the sinks of H. $\square$

In the literature, a "complete" pebbling of G is sometimes defined as a pebbling that touches all vertices of G. We note that this is equivalent to the definition in this paper.

Observation 2.3. *$\mathcal{P}$ is a complete pebbling of G (of any type) in the sense that it pebbles all sinks if and only if $\mathcal{P}$ places a pebble on each vertex in G at least once.*

Proof. The if-direction is true by definition. To see that the only-if-direction also holds, suppose not and pick some v at minimum distance from any of the sinks z such that v is never pebbled but all other vertices on the shortest path $Q : v \rightsquigarrow z$ are pebbled. But then the successor of v on this path must have been pebbled in violation of the pebbling rules. $\square$

Some papers on pebbling deal with a slightly different flavour of the game, where it is also allowed to slide pebbles.

Definition 2.4 (Sliding). Suppose that G is a DAG with sources S and sinks Z. A pebbling of G with *sliding* is a pebbling according to the rules in Definition 1.1 plus the following two rules:

5. If all immediate predecessors of an empty vertex v have pebbles on them and w is a predecessor with a black pebble, then the black pebble on w may be slid from w to v.

6. If all immediate predecessors of a white-pebbled vertex v except one vertex w have pebbles on them, then the white pebble on v may be slid to w.

The relation between pebbling with and without sliding is spelled out in the next proposition.

Proposition 2.5 ([EBL79, GLT80]). *Let $\mathcal{P}$ be a pebbling strategy of any type for G using sliding. Then there is a strategy $\mathcal{P}'$ of the same type but without sliding such that $\mathsf{time}(\mathcal{P}') \leq 2 \cdot \mathsf{time}(\mathcal{P})$ and $\mathsf{space}(\mathcal{P}') \leq \mathsf{space}(\mathcal{P}) + 1$.*

Moreover, the black pebbling prices of G with and without sliding differ by exactly 1, and if G is a single-sink DAG, the black-white pebbling prices for G with and without sliding differ by exactly 1.

Proof. The upper bounds for $\mathcal{P}'$ in terms of $\mathcal{P}$ are obvious—just replace every sliding by a pair of pebble placements and removals.

To see that sliding can always save one pebble for single-sink DAGs, consider a time t when the maximal space is reached in a pebbling without sliding by a pebble placement on some vertex v. We make a case analysis over the move at time $t + 1$ (which must be a pebble removal) to see how a sliding pebbling can save space.

1. Pebble removal from $u \in \mathit{pred}(v)$: if so, the pebble on u is (or can be made) black, and we can slide this pebble from u to v instead.

2. Removal of the pebble on v: If v is not the sink, we can omit the placement and immediate removal of a pebble on v without affecting the rest of the pebbling in any way, so suppose v is the sink. If there is a predecessor u of v with no other successors than v and with a black pebble, slide this pebble to v instead and then remove it. The pebble on u will never have to be used again since its only successor has now been pebbled. Otherwise, there is a white-pebbled vertex u with the sink v as its only successor. The white pebble on u cannot have been needed in the pebbling before time t, so we can omit the move placing the pebble there and instead slide a white pebble from v down to u at time t.

3. Removal of some pebble not on $\{v\} \cup \mathit{pred}(v)$: Just switch the order of the moves and perform this removal at time t before placing the pebble on v at time $t + 1$.

In a black-only pebbling of a multi-sink DAG, after having slid a pebble to the sink in case 2 we can restart the whole pebbling from scratch and run it up to time $t - 1$, ignore the sink pebble placement at time t, and then continue with the rest of the pebbling. □

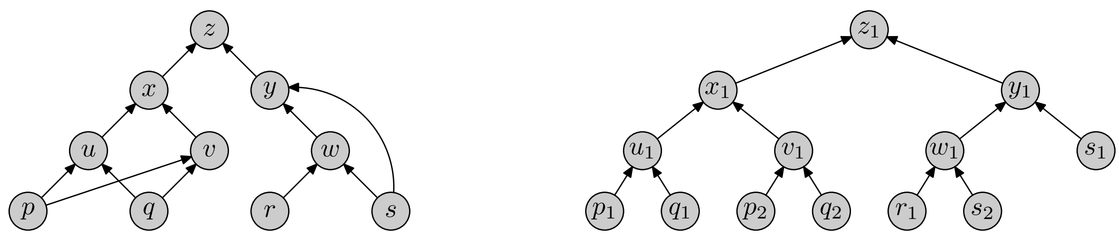
For the purposes of this survey, the multiplicative factor of 2 for the time and additive term 1 for the space in Proposition 2.5 is of no consequence, and therefore we will focus entirely on pebbling strategies *without* sliding. We note however, that if one is interested in the exact number of moves needed to obtain a pebbling in minimal space, the one pebble saved by using the sliding rule can incur a quadratic time penalty in multi-sink DAGs. See [EBL79] for more details.

3 Some Pebbling Technicalities

We continue our discussions of preliminaries started in Section 2, but now turn our attention to some technical definitions and observations that will come in handy in the proofs. While we have tried to attribute all definitions and results below to the correct papers, it should be mentioned that the citations are somewhat arbitary in the sense than many of these technical tools appear to have been independently reinvented several times.

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

Observation 3.1 (Trivial pebbling). *Any DAG G can be completely, persistently black-pebbled in space at most $|V(G)|$ and time at most $2 \cdot |V(G)|$ simultaneously.*



(a) General graph G of depth 3. (b) Unfolding of G into the tree $\mathit{unfold}(G)$.

Figure 3: Example of unfolding of a graph.

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

Definition 3.2 (Depth). The *depth* of a DAG G is the length of a longest path from a source to a sink in G.

Observation 3.3. *Any DAG G with maximal indegree ℓ and depth d has a black pebbling strategy in space at most $d\ell + 1$.*

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

The proof of of Observation 3.3 can be seen to pebble each vertex v as if $G^v_\triangle$ was a tree disjoint from the rest of G. It will be convenient to formalize this idea of "unfolding" a graph to a tree and then using a pebbling strategy for this tree to pebble G.

Definition 3.4 (Unfolding ([LT82])). For any DAG G with sinks $Z(G) = \{z_1, \ldots, z_m\}$, the *unfolding* of G, which we denote $\mathit{unfold}(G)$, is the tree defined recursively as follows.

The sinks of $\mathit{unfold}(G)$ are $\{z_1, \ldots, z_m\}$. If the first sink z_1 in G has predecessors x and y, we create two predecessors x_1 and y_1 of z_1 in $\mathit{unfold}(G)$. Recursively, for any vertex in $\mathit{unfold}(G)$ labelled by, say, w_i, look at the corresponding vertex w in G and suppose for concreteness that $\mathit{pred}(w) = \{u, v\}$. Then create new vertices $\mathit{pred}(w_i)$ in $\mathit{unfold}(G)$ labelled by u_j and v_k for the smallest positive indices j, k such that there are not already other vertices in $\mathit{unfold}(G)$ labelled u_j and v_k.

Definition 3.4 is illustrated in Figure 3. We have the following easy proposition.

Proposition 3.5 ([LT82]). *Any complete pebbling $\mathcal{P}$ of $\mathit{unfold}(G)$ induces a complete pebbling $\mathcal{P}'$ of G such that $\mathsf{time}(\mathcal{P}') \leq \mathsf{time}(\mathcal{P})$ and $\mathsf{space}(\mathcal{P}') \leq \mathsf{space}(\mathcal{P})$.*

Proof. Given any pebbling strategy $\mathcal{P}$ for $\mathit{unfold}(G)$, we can pebble G with at most the same amount of pebbles by mimicking any move on any v_i in $\mathit{unfold}(G)$ by performing the same move on v in G. The details are easily verified. $\square$

Some proofs are facilitated by observing that visiting pebblings have a certain "duality" property. The next proposition is an immediate consequence of the anti-symmetric nature of the pebbling rules in Definition 1.1 (just observe that the rules for placing and removing a black pebble are the duals of the rules for removing and placing a white pebble, respectively).

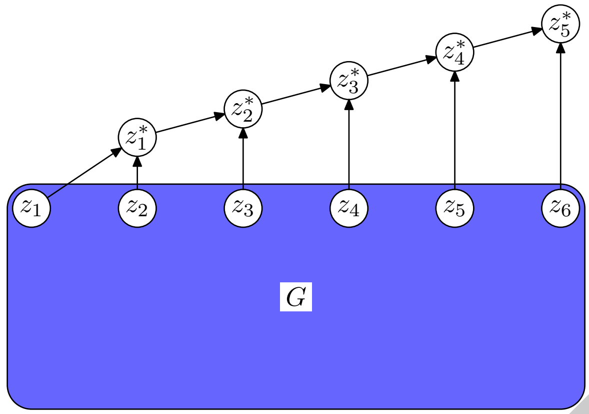


Figure 4: Schematic illustration of single-sink version $\widehat{G}$ of graph G.

Proposition 3.6 ([CS76]). *Suppose that $\mathcal{P}$ is a black-white pebbling from (B_1, W_1) to (B_2, W_2). Then we can get a dual pebbling $\overline{\mathcal{P}}$ from (W_2, B_2) to (W_1, B_1) in exactly the same cost by reversing the sequence of moves and switching the colours of the pebbles. In particular, if $\mathcal{P}$ is a complete visiting pebbling of G, then so is $\overline{\mathcal{P}}$.*

For the applications in proof complexity, we often want results stated for DAGs with one unique sink, but most pebbling results are more natural to state and prove for DAGs with multiple sinks. This small technicality is easily taken care of. We do this next.

Definition 3.7 (Single-sink version). Let G be a DAG with sinks $Z(G) = \{z_1, \ldots, z_m\}$ for $m > 1$. The *single-sink version* $\widehat{G}$ of G consists of all vertices and edges in G plus the extra vertices $z_1^*, \ldots, z_{m-1}^*$ and the edges (z_1, z_1^*), (z_2, z_1^*), (z_1^*, z_2^*), (z_3, z_2^*), (z_2^*, z_3^*), (z_4, z_3^*), et cetera up to (z_{m-2}^*, z_{m-1}^*), (z_m, z_{m-1}^*).

That is, $\widehat{G}$ consists of G with a binary tree of minimal size added on top of the sinks. See Figure 4 for a picture of this. The following observation is immediate.

Observation 3.8. *Let G be a DAG with sinks $Z(G) = \{z_1, \ldots, z_m\}$ for $m > 1$. Then for any flavour of pebbling (visiting or persistent) it holds that $\mathsf{BW\text{-}Peb}(\widehat{G}) \leq \mathsf{BW\text{-}Peb}(G) + 1$ and $\mathsf{Peb}(\widehat{G}) \leq \mathsf{Peb}(G) + 1$. Moreover, if there is a pebbling strategy $\mathcal{P}$ (visiting or persistent) for G in space s that can pebble the sinks in arbitrary order, then there is a pebbling strategy $\widehat{\mathcal{P}}$ of the same type (black or black-white, visiting or persistent) for $\widehat{G}$ with $\mathsf{time}(\widehat{\mathcal{P}}) \leq \mathsf{time}(\mathcal{P}) + 2m$ and $\mathsf{space}(\widehat{\mathcal{P}}) \leq \mathsf{space}(\mathcal{P}) + 1$.*

The next proposition will be used a number of time when composing pebblings of smaller subgraphs into a pebbling of a larger graph.

Proposition 3.9. *Suppose that G is a DAG with unique sink z. Then for any complete black or black-white pebbling $\mathcal{P}$ of G in space s there is a complete pebbling $\mathcal{P}'$ with the same colours such that $\mathsf{time}(\mathcal{P}') = \mathsf{time}(\mathcal{P})$, $\mathsf{space}(\mathcal{P}') = \mathsf{space}(\mathcal{P})$, and there is a time t during $\mathcal{P}'$ when z has a pebble but the pebbling space is strictly less than s.*

Proof. For black pebblings this statement is obvious. Once we place a black pebble on the sink z, we can remove all other pebbles from the DAG.

Suppose for a black-white pebbling $\mathcal{P}$ that the pebbling space reaches the maximum s precisely when a pebble is placed on z at time t. Then the move at time $t + 1$ must be a pebble removal. If a pebble is removed from another vertex, we are done. Otherwise, fix some vertex $w \in \mathit{pred}(z)$ having z as its only successor. Suppose that w contains a white pebble during some interval $[\sigma, \tau] \supseteq [t, t + 1]$ (and if not, run the dual pebbling in Proposition 3.6 instead). To obtain $\mathcal{P}'$, we change $\mathcal{P}$ as follows. The pebble placement on w at time σ is omitted. At time t, a white pebble is placed on z. In between times t and $t+1$, w is white pebbled, and then the white pebble on z is removed at time $t+1$. □

It is immediate from the definition of the black pebble game that black pebblings always proceed in a bottom-up fashion in the following sense.

Observation 3.10. *Suppose that $Q : u \rightsquigarrow v$ is a path in G and that $\mathcal{P} = \{\mathbb{P}_\sigma, \mathbb{P}_{\sigma+1}, \ldots, \mathbb{P}_\tau\}$ is a black-only pebbling such that the whole path Q is completely free of pebbles at time σ but a pebble is placed on the endpoint v at time τ. Then the starting point u must have been pebbled during the time interval (σ, τ).*

A simple but important lemma that lies at the heart of most black-white pebbling lower bounds is the following generalization of Observation 3.10 to black-white pebbling: In order to pebble the endpoint v of some path, one needs to pebble all vertices on this path at some point prior to *or after* pebbling v.

Lemma 3.11 ([GT78]). *Suppose that $Q : u \rightsquigarrow v$ is a path in G and that $\mathcal{P} = \{\mathbb{P}_\sigma, \mathbb{P}_{\sigma+1}, \ldots, \mathbb{P}_\tau\}$ is a black-white pebbling such that the whole path Q is completely free of pebbles at times σ and τ but the endpoint v is pebbled at some point in the time interval (σ, τ). Then the starting point u is pebbled during (σ, τ) as well.*

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

When proving lower bounds on pebblings, it often helps to assume that the pebblings under consideration do not perform any obviously redundant moves. The following definition, which formalizes this notion, is a generalization of [GLT80] from black-only to black-white pebbling.

Definition 3.12 (Frugal pebbling). Let $\mathcal{P}$ be a complete pebbling of a DAG G. To every pebble placement on a vertex v at time σ we associate the *pebbling interval* $[\sigma, \tau)$, where $\tau = \tau(\sigma, v)$ is the first time after σ when the pebble is removed from v again (or $\tau = \infty$, say, if this never happens).

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

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

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

Lemma 3.13. *For any complete pebbling $\mathcal{P}$ (black or black-white, visiting or persistent) there is a frugal pebbling $\mathcal{P}'$ of the same type such that $\mathsf{time}(\mathcal{P}') \leq \mathsf{time}(\mathcal{P})$ and $\mathsf{space}(\mathcal{P}') \leq \mathsf{space}(\mathcal{P})$.*

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

One minor technical snag is that we will need to assume not only that complete pebblings are frugal, but that this also holds for *conditional pebblings* (Definition 2.1). This is no real problem, however, since we can always assume that the conditional pebblings we are dealing with are subpebblings of some larger, unconditional pebbling. In fact, an alternative way of defining frugal pebblings (unconditional or conditional) is to say that a pebble placement on a non-sink vertex v is essential if the pebble stays until either a black pebble is placed on an immediate successor of v or a white pebble is removed from an immediate successor of v. If a pebbling contains non-essential moves, then it is easy to see that such moves can be eliminated to get a shorter pebbling that is still legal. This new pebbling might contain other non-essential moves, but after applying the procedure a finite number of times we obtain a pebbling with only essential moves. Adding the requirement that each sink should only be pebbled once, we recover Definition 3.12.

4 Layered Graphs

In this section, we study graphs that can be decomposed into layers with edges going between consecutive layers and give upper and lower bounds on the pebbling price of such graphs.

Perhaps the simplest example of layered graphs, and certainly the simplest graphs to analyze, 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 proven independently in [Lou79, Mey79]. We state the bound only for binary trees since these are the trees we will be interested in in this survey.

Theorem 4.1 ([Lou79, Mey79]). *For the complete binary tree T_h of height $h \geq 1$ it holds that $\mathsf{Peb}(T_h) = h+2$, $\mathsf{BW\text{-}Peb}^{\bullet}(T_h) = \lfloor \frac{h}{2} \rfloor + 3$, and $\mathsf{BW\text{-}Peb}^{\emptyset}(T_h) = \lfloor \frac{h-1}{2} \rfloor + 3$.*

Thus, for complete trees one can save roughly half of the pebbles by using black and white pebbles instead of black pebbles only. It was shown in [LT80] that one can never save more than half of the pebbles for any (non-complete) tree of any arity, but it is not hard to produce degenerate trees for which the black and black-white pebbling prices coincide (for instance, trees that are essentially collections of lines).

Another class of layered DAGs that will be of particular interest to us are so-called pyramid graphs. We have not defined pyramid graphs formally yet (this will be done in Definition 4.5 below) but hopefully it is clear from the example pyramid of height 2 in Figure 1 what these graphs look like.

Theorem 4.2 ([Coo74, Mey81, Kla85]). *The black pebbling price of the pyramid graph Π_h of height h is $\mathsf{Peb}(\Pi_h) = h+2$ and there is a linear-time pebbling achieving this bound.*

The black-white pebbling price is $\mathsf{BW\text{-}Peb}(\Pi_h) = h/2 + \mathrm{O}(1)$. For pyramids of odd height the exact bound $\mathsf{BW\text{-}Peb}^{\bullet}(\Pi_{2h+1}) = h+3$ holds, and for even height we have $\mathsf{BW\text{-}Peb}^{\emptyset}(\Pi_{2h}) = h+2$.

The lower bound for black pebbling in Theorem 4.2 is from Cook [Coo74], and it is easy to construct a linear-time pebbling matching this bound by pebbling the pyramid bottom-up, layer by layer, or simply by noting that unfolding the pyramid (Definition 3.4) we get a complete binary tree and then apply Proposition 3.5 (compare with Lemma 4.7 below). In the same way, the black-white pebbling strategy for pyramids in space $h/2 + \mathrm{O}(1)$ can be obtained from the corresponding strategy for binary trees but (apparently before this was noted) Meyer auf der Heide [Mey81] gave a black-white strategy specifically for pyramids which is slightly more efficient with respect to time. Later, Klawe [Kla85] showed that $h/2 + \mathrm{O}(1)$ is also a lower bound on the black-white pebbling price, improving on a previous bound $\Omega(\sqrt{h})$ by Cook and Sethi [CS76] and thus resolving a relatively long-standing open problem. The exact bounds for black-white pebbling stated in Theorem 4.2 were not proven in [Kla85], however, but can be found in the exposition of Klawe's construction in [NH08b] (or rather in the full-length version [NH08a] of that paper).

4.1 Definitions, Terminology and Notation for Layered DAGs

To prove Theorems 4.1 and 4.2, we first set up some notation and terminology.

Definition 4.3 (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.

For the layered DAGs G that we will study in this paper, 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. This means that the height will be equal to the depth as defined in Definition 3.2, but we will stick with the term of "height" for layered DAGs, reserving "depth" for DAGs that might not be layered.

The following notation will be convenient.

Definition 4.4 (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 completely analogously.

Let us next give the formal definition of pyramids.

Definition 4.5 (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 $L-1$.

Although most of what will be said in what follows holds for arbitrary layered DAGs, in this section we will tend to focus on pyramids. Figure 5(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 (viewing pyramids in this way might help the intuition in some of the proofs). In Figure 5(b), the pyramid in Figure 5(a) is redrawn as such a lattice fragment.

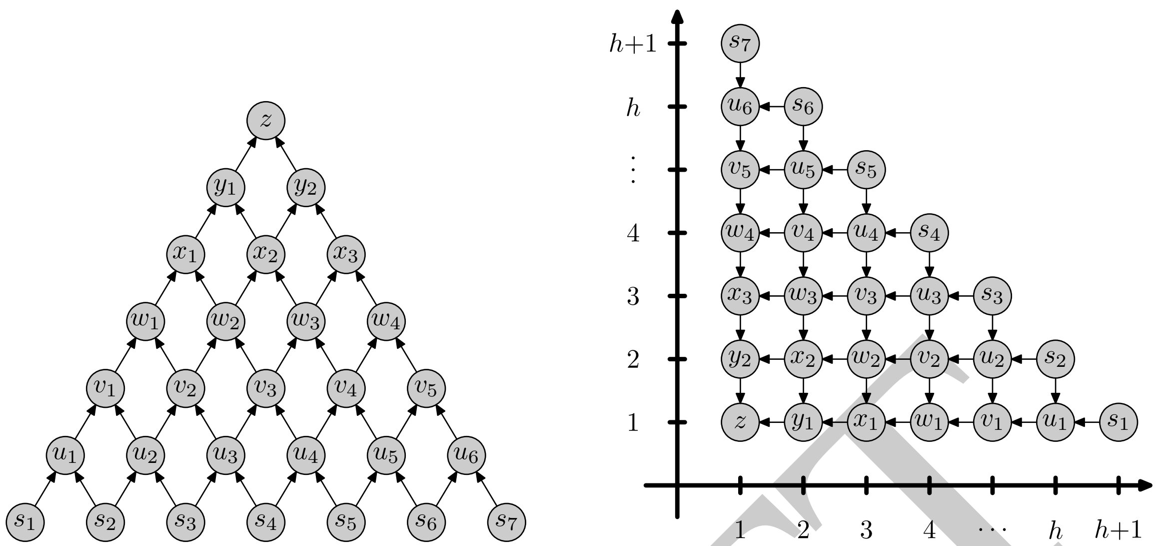
We also need some notation for contiguous and non-contiguous topologically ordered sets of vertices in a DAG. Recall that we write $P : v \rightsquigarrow w$ to denote a path starting in v and ending in w.

Definition 4.6 (Chain). We say that V is a *(totally) ordered* set of vertices in a DAG G, or a *chain*, if all vertices in V are comparable (i.e., if for all $u, v \in V$, either $u \in G^v_\Delta$ or $v \in G^u_\Delta$ holds). Thus, 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. For a chain V, we let

- $\mathrm{bot}(V)$ denote the bottom vertex of V, i.e., the unique $v \in V$ such that $V \subseteq G^\nabla_v$,
- $\mathrm{top}(V)$ denote the top vertex of V, i.e., the unique $v \in V$ such that $V \subseteq G^v_\Delta$,
- $\mathfrak{P}_{\mathrm{in}}(V)$ 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
- $\mathfrak{P}_{\mathrm{via}}(V)$ denote the set of all source paths *agreeing with* V.

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

Unless otherwise stated, in the rest of this section 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.



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

(b) Pyramid as fragment of 2D lattice.

Figure 5: Running example pyramid Π_6 of height 6 with labelled vertices.

4.2 Pebbling Price of Binary Trees

Recall that T_h denotes the complete binary tree of height h considered as a DAG with edges directed towards the root. The fact that $\mathit{Peb}(T_h) = h + 2$ can be established by induction over the tree height. We omit the easy proof. It remains to establish the claims made in Theorem 4.1 with regard to black-white pebbling.

Proof of Theorem 4.1. Throughout this proof, we let z_1, z_2 denote the immediate predecessors of the root z of the tree.

We first show that $\mathit{BW\text{-}Peb}^{\emptyset}(T_{h+2}) \geq \mathit{BW\text{-}Peb}^{\emptyset}(T_h) + 1$. Suppose not, and let $\mathcal{P}$ be a pebbling in space $K = \mathit{BW\text{-}Peb}^{\emptyset}(T_h)$ for T_{h+2} making the minimum number of pebbling moves. Let $T_h^{(i)}$, $i \in [4]$, be the four disjoint subtrees of height h in T_{h+2}. Since $\mathcal{P}$ restricted to $T_h^{(i)}$ yields a visiting pebbling of $T_h^{(i)}$ (Observation 2.2), it follows that there must exist distinct times t_i, $i \in [4]$, when $T_h^{(i)}$ contains K pebbles and the rest of T_{h+2} is empty. Number the subtrees so that $t_1 < t_2 < t_3 < t_4$.

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

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

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

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

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

$\lfloor \frac{h+1}{2} \rfloor + 3$. In the same way we get $\textsf{\textit{BW-Peb}}^\bullet(T_{h+1}) = \lfloor \frac{h+1}{2} \rfloor + 3$.

To pebble T_{h+2} in space $\lfloor \frac{h+1}{2} \rfloor + 3$ leaving a pebble on z, first black-pebble z_1, which we recall is the root of a subtree of height $h+1$, in space $\lfloor \frac{h+1}{2} \rfloor + 3$. Leaving the pebble on z_1, make a pebbling visiting z_2 in space $\lfloor \frac{h}{2} \rfloor + 3 = \lfloor \frac{h+1}{2} \rfloor + 2$ using the pebbling for T_{h+1} constructed above. In this pebbling there is a time t when z_2 is pebbled and the subtree rooted at z_2 contains at most $\lfloor \frac{h+1}{2} \rfloor + 1$ pebbles (by Proposition 3.9). At this time t, place a black pebble on z and remove the black pebble on z_1 without exceeding the total limit of $\lfloor \frac{h+1}{2} \rfloor + 3$ pebbles on T_{h+2}. Then finish the pebbling. The theorem follows. □

We can use our knowledge of the black and black-white pebbling price of binary trees to get upper bounds on pebbling price for any layered DAG.

Lemma 4.7. *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 $\textsf{\textit{Peb}}(G_h) \leq h + \mathrm{O}(1)$ and $\textsf{\textit{BW-Peb}}^\bullet(G_h) \leq h/2 + \mathrm{O}(1)$.*

Proof. The bounds above are true for complete binary trees of height h, as we have just seen. If we take a layered DAG G_h of height h and indegree 2 and unfold it (Definition 3.4) we get a binary tree of height h. We can then use the pebbling strategies for binary trees to pebble G_h as well (Proposition 3.5). □

4.3 The Black Pebbling Price of Pyramids

The purpose of the rest of this section is to identify some layered graphs G_h for which the bound in Lemma 4.7 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 pyramids.

Theorem 4.8 ([Coo74]). *For pyramids Π_h of height $h \geq 1$ it holds that $\textsf{\textit{Peb}}(\Pi_h) = h + 2$, and there is a linear-time complete black pebbling achieving this space bound.*

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

Definition 4.9 (Blocking). 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 4.8. The fact that $\textsf{\textit{Peb}}(\Pi_h) \leq h + 2$ follows from Lemma 4.7, but the pebbling obtained from this lemma takes exponential time. If we instead black-pebble the pyramid bottom-up by first placing pebbles on all $h + 1$ sources and then, using one auxiliary pebble, move these pebbles up to level 1, 2, et cetera all the way up to the sink, the pebbling time is linear.

We now show that $h + 2$ 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. Note that P must be blocked by a pebble placement on some source vertex u, since otherwise both vertices in $\mathit{pred}(u)$ would have to have pebbles on them and so P would already be blocked. 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 repeatedly the idea in the proof above about a set of paths converging at different levels to another fixed path, so we write it down as a separate observation.

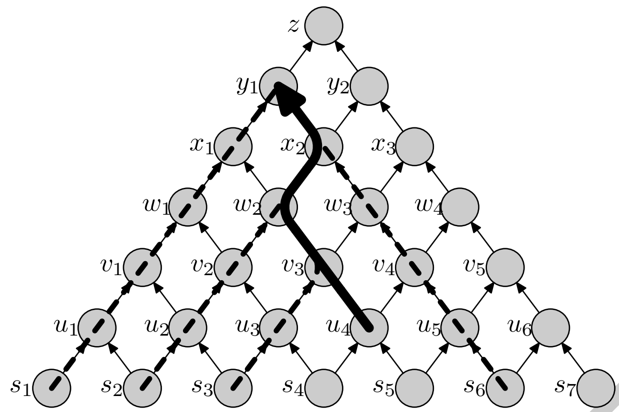


Figure 6: Set of converging source paths (dashed) for the path $P : u_4 \rightsquigarrow y_1$ (solid).

Observation 4.10. *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 but 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 6 for an example.

4.4 Black-White Pebbling Pyramids—a First Bound

For the black-white pebble game, Cook and Sethi proved the following lower bound on the pebbling price of pyramids.

Theorem 4.11 ([CS76]). *BW-Peb*$^\bullet(\Pi_h) \geq \frac{1}{2}\sqrt{h}$.

In this subsection, we give a rather detailed exposition of the proof of this theorem, in the hope that this will provide helpful intuition for the tighter (and more intricate) lower bound proof that will follow next. Cook and Sethi get their bound by proving something slightly stronger than in the statement of Theorem 4.11, namely the following.

Lemma 4.12 ([CS76]). *Suppose that $(B_0, \emptyset)$ and $(\{z\}, W_\tau)$ are pebble configurations in a pyramid Π_h such that there is a path $P : v \rightsquigarrow z$ from a source vertex v to the sink z with $P \cap (B_0 \cup W_\tau) = \emptyset$. Then for any conditional pebbling $\mathcal{P}$ from $(B_0, \emptyset)$ to $(\{z\}, W_\tau)$ it holds that* $\mathsf{space}(\mathcal{P}) \geq \frac{1}{2}\sqrt{h}$.

That is, we start with a possibly non-empty set of black pebbles on B_0 and end with a possibly non-empty set of white pebbles on W_τ, but we have somehow managed to place a black pebble on the sink z and empty a path from a source to z that was not blocked when we started and is not blocked when we end. It is clear that Theorem 4.11 follows from this by taking $B_0 = W_\tau = \emptyset$.

In order to prove Lemma 4.12, we will use the anti-symmetry property of the black-white pebble game in Proposition 3.6 again. Note that, in particular, Proposition 3.6 implies that if B_0 and W_τ are such that there is a path $P : v \rightsquigarrow z$ from a source vertex v to the sink z with $P \cap (B_0 \cup W_\tau) = \emptyset$, then the minimal space of any pebbling from $(B_0, \emptyset)$ to $(\{z\}, W_\tau)$, i.e., pebblings *placing a black pebble* on the sink of the pyramid, will be equal to the minimal space of any pebbling from $(B_0, \{z\})$ to $(\emptyset, W_\tau)$, i.e., pebblings *removing a white pebble* from the sink of the pyramid.

Proof of Lemma 4.12. The proof is by induction over the height h of the pyramid Π. For the induction base, note that it is obvious for any pyramid of height $h \geq 1$ that $\mathit{space}(\mathcal{P}) \geq 3$. The sink z has a black pebble at time τ. At the time when this pebble was placed on z, both its predecessors must also have had pebbles on them. Thus for $h \leq 36$ we have $\mathit{space}(\mathcal{P}) \geq \frac{1}{2}\sqrt{h}$.

Suppose that the statement in the theorem is true for all $h' < h$. By the anti-symmetry in Proposition 3.6 we can assume that the bound holds for *all* pebblings $\mathcal{P}$ such that there is a path $P : v \rightsquigarrow z$ from a source vertex v to the sink z with $P \cap (B_0 \cup W_\tau) = \emptyset$, and such that $\mathcal{P}$ either leaves a black pebble on z or removes an initial white pebble from z.

Now we do the induction step. Suppose for a pyramid Π of height h that $\mathcal{P}$ is a pebbling in minimum space from $(B_0, \emptyset)$ to $(\{z\}, W_\tau)$. Without loss of generality, we can assume that $\mathcal{P}$ is a pebbling with the least number of pebbling moves among such minimum-space pebblings.

Consider the last time σ when we have a configuration (B_σ, W_σ) such that $B_\sigma \cup W_\tau$ blocks all paths from sources to z, but this is not true for $B_{\sigma-1} \cup W_\tau$. Clearly, there is such time σ since $B_0 \cup W_\tau$ does not block all paths to z but $B_\tau \cup W_\tau = \{z\} \cup W_\tau$ trivially does.

Since $B_{\sigma-1} \cup W_\tau$ does not block all paths from sources to z, the move at time σ must be a placement of a black pebble on some vertex r. Also, if r is not a source there must exist at least one white-pebbled predecessor q of r and a path P from a source via q and r to z such that P is not blocked by $B_{\sigma-1} \cup W_\tau$. (Both predecessors must have pebbles at time $\sigma-1$, but if both predecessors were black-pebbled, $B_{\sigma-1} \cup W_\tau$ would already block all paths.)

We claim that if r is at distance $2\sqrt{h}$ or more from the sink, then we are done. For consider the converging paths of Observation 4.10 for the subpath $P^r : r \rightsquigarrow z$ of P. All these paths must be blocked by $(B_\sigma \cup W_\tau) \setminus \{r\} = B_{\sigma-1} \cup W_\tau$ but there are no pebbles from $B_{\sigma-1} \cup W_\tau$ on P^r, so $|B_{\sigma-1} \cup W_\tau| \geq 2\sqrt{h}$ yielding a pebbling space of at least $\max\{|B_{\sigma-1}|, |W_\tau|\} \geq \sqrt{h}$. Suppose therefore that r is at distance strictly less than $2\sqrt{h}$ from the sink.

Consider the subpyramid Π^q_Δ rooted at q. The height of Π^q_Δ is at least $h - 2\sqrt{h}$. At time σ there is a white pebble at the sink q and a path P' from a source to q (namely the subpath $P' = P \cap \Pi^q_\Delta$ of P) such that $B_\sigma \cup W_\tau$ does not block P'. Looking just at the pebbles inside Π^q_Δ during the time interval $[\sigma, \tau]$ and using Observation 2.2, we see that we get a pebbling removing the white pebble on q and opening a path to q in Π^q_Δ. By the induction hypothesis (and anti-symmetry), this pebbling inside Π^q_Δ costs at least $\frac{1}{2}\sqrt{h - 2\sqrt{h}} \geq \frac{1}{2}\sqrt{h} - 1$ if $h \geq 4$.

Also, we claim that during the whole time interval $[\sigma, \tau]$ there must be a pebble in Π outside Π^q_Δ. This is true at time σ and at time τ. Suppose there is some time $t \in (\sigma, \tau)$ such that all pebbles are inside Π^q_Δ. Then it is easy to verify that we get a correct pebbling $\mathcal{P}'$ of all of Π by ignoring all pebble placements and removals outside Π^q_Δ before time t. This pebbling $\mathcal{P}'$ has at most the same space as $\mathcal{P}$ and has strictly fewer moves (since it does not place a black pebble on r at time σ, for instance), contradicting the assumed minimality of $\mathcal{P}$. Thus there is at least one pebble outside Π^q_Δ during the whole time interval $[\sigma, \tau]$, so the total cost of $\mathcal{P}$ is at least $1 + (\frac{1}{2}\sqrt{h} - 1) = \frac{1}{2}\sqrt{h}$. The theorem follows. □

4.5 A Tight Bound for Black-White Pebbling Layered DAGs

Reading the proof of Lemma 4.12, it somehow seems unlikely that the analysis can be tight. It took quite some time and effort, however, before Klawe [Kla85] managed to remove the square root from the lower bound. In the remainder of this section, we give a detailed exposition of the lower bound in [Kla85], somewhat simplifying the proof and obtaining almost exact bounds. Much of the notation and terminology has been changed from [Kla85] to fit better with the rest of this paper. 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 [Kla85] can give a better size/pebbling price trade-off than one gets for layered graphs (which is what we want for the proof complexity applications that we have in mind).

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

Definition 4.13 (Hiding set). A vertex set U *hides* a vertex w if U blocks all source paths visiting w, i.e., if U blocks $\mathfrak{P}_{\mathrm{via}}(w)$. U hides W if U hides all $w \in W$. If so, we say that U is a *hiding set* for W. We write $\llbracket U \rrbracket$ to denote the set of all vertices hidden by U.

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 4.14. *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 $\mathsf{space}(\mathbb{P})$. 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 space 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 4.15 (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 4.16 (Potential). We say that U is a hiding set for a black-white pebble configuration $\mathbb{P} = (B, W)$ in a layered graph G if $U \cup W$ hides B. We define the *potential* of the pebble configuration to be

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

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

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

We remark that although this might not be immediately obvious, there is quite a lot of nice intuition why Definition 4.16 is a relevant estimation of how "good" a pebble configuration is. We refer the reader to Section 2 of [Kla85] for a discussion of 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.

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

1. The potential correctly estimates the goodness of the current configuration $\mathbb{P}_t$ by taking into account the whole pebbling that has led to $\mathbb{P}_t$. Namely, $\mathrm{pot}(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathit{space}(\mathbb{P}_s)\}$.

2. The final configuration $\mathbb{P}_\tau = (\{z\}, \emptyset)$ has high potential, namely $\mathrm{pot}(\{z\}, \emptyset) = h + \mathrm{O}(1)$.

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

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

Property 4.17 (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. $\mathrm{pot}_G(\mathbb{P}) = m_G(U)$,

3. $U = B$ or $|U| < |B| + |W| = \mathit{space}(\mathbb{P})$.

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 space 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 $\mathit{space}(\mathbb{P})$.

Given Property 4.17, the induction proof for part 1 follows quite easily. The main part of the paper [Kla85] is then spent on proving that a class of DAGs including pyramids have Property 4.17. Let us see what the lower bound proof looks like, assuming that Property 4.17 holds.

Lemma 4.18 (Theorem 2.2 in [Kla85]). *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 $\mathrm{pot}_G(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathit{space}(\mathbb{P}_s)\}$.*

Proof. The proof is by induction. The base case $\mathbb{P}_0 = \emptyset$ is trivial. For the induction hypothesis, suppose that $\mathrm{pot}(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathit{space}(\mathbb{P}_s)\}$ and let U_t be a vertex set as in Property 4.17, i.e., such that $U_t \cup W_t$ hides B_t, $\mathrm{pot}(\mathbb{P}_t) = m(U_t)$ and $|U_t| \leq \mathit{space}(\mathbb{P}_t) = |B| + |W|$.

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

$$\mathrm{pot}(\mathbb{P}_{t+1}) \leq \max\{\mathrm{pot}(\mathbb{P}_t), 2 \cdot \mathit{space}(\mathbb{P}_{t+1})\} \ . \tag{4.1}$$

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

   The case when v is a source is more interesting. 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 $\operatorname{pot}(\mathbb{P}_{t+1}) = m(U_t \cup \{v\}) > m(U_t) = \operatorname{pot}(\mathbb{P}_t)$. And we do not know that the inequality $\operatorname{pot}(\mathbb{P}_t) \leq 2 \cdot \mathsf{space}(\mathbb{P}_t)$ holds, only that $\operatorname{pot}(\mathbb{P}_t) \leq 2 \cdot \max_{s \leq t}\{\mathsf{space}(\mathbb{P}_s)\}$. This means that it can happen that $\operatorname{pot}(\mathbb{P}_{t+1}) > 2 \cdot \mathsf{space}(\mathbb{P}_{t+1})$, in which case the induction step fails. However, we claim that using the Limited hiding-cardinality property 4.17 we can prove for $U_{t+1} = U_t \cup \{v\}$ that

   $$m(U_{t+1}) = m(U_t \cup \{v\}) \leq \max\{m(U_t), 2 \cdot \mathsf{space}(\mathbb{P}_{t+1})\} \ , \tag{4.2}$$

   which shows that (4.1) holds and the induction steps goes through.

   Namely, suppose that U_t is chosen as in Property 4.17 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 $\operatorname{pot}(\mathbb{P}_{t+1}) \leq m(U_{t+1})$. 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 $|U_t| \leq \mathsf{space}(\mathbb{P}_t)$ holds by the LHC property, we have

   $$m^0(U_{t+1}) = 2 \cdot |U_{t+1}| = 2 \cdot (|U_t| + 1) \leq 2 \cdot (\mathsf{space}(\mathbb{P}_t) + 1) = 2 \cdot \mathsf{space}(\mathbb{P}_{t+1}) \tag{4.3}$$

   and we get that

   $$\begin{aligned} m(U_{t+1}) = \max_j\{m^j(U_{t+1})\} &= \max\{\max_{j>0}\{m^j(U_t)\}, m^0(U_{t+1})\} \\ &\leq \max\{m(U_t), 2 \cdot \mathsf{space}(\mathbb{P}_{t+1})\} = \max\{\operatorname{pot}(\mathbb{P}_t), 2 \cdot \mathsf{space}(\mathbb{P}_{t+1})\} \end{aligned} \tag{4.4}$$

   which is exactly what we need.

4. Removal of white pebble: Suppose that a white pebble is removed from the vertex w, so $W_{t+1} = W_t \setminus \{w\}$. We claim that if w is not a source vertex, then $U_t \cup W_{t+1}$ still hides $B_{t+1} = B_t$, from which $\operatorname{pot}(\mathbb{P}_{t+1}) \leq \operatorname{pot}(\mathbb{P}_t)$ 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 4.14. 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}.

   If w is a source vertex, the argument is similar to that in case 3 although here it is slightly more subtle. Let the hiding set U_t for (B_t, W_t) be chosen so that Property 4.17 holds. If U_t does not hide (B_{t+1}, W_{t+1}), then, in particular, $U_t \neq B_t = B_{t+1}$ and it follows from Property 4.17 that $|U_t| < \mathsf{space}(\mathbb{P}_t)$. Let us again choose $U_{t+1} = U_t \cup \{v\}$ to hide $\mathbb{P}_{t+1}$. Then as in case 3 it holds that $m^j(U_{t+1}) = m^j(U_t)$ for $j > 0$, and the only thing that remains to bound the potential is to study the partial measure at the bottom level. Here we have

   $$m^0(U_{t+1}) = 2 \cdot (|U_t| + 1) \leq 2 \cdot \mathsf{space}(\mathbb{P}_t) \tag{4.5}$$

   from which we obtain the bound

   $$\operatorname{pot}(\mathbb{P}_{t+1}) \leq \max\{\operatorname{pot}(\mathbb{P}_t), 2 \cdot \mathsf{space}(\mathbb{P}_t)\} \leq 2 \cdot \max_{s \leq t}\{\mathsf{space}(\mathbb{P}_s)\} \ . \tag{4.6}$$

We see that the inequality (4.1) 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 4.19. *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 $m(U) = m^h(U) = h + 2$. 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 $L < h$. 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 4.10, there must exist $h - L$ converging paths from sources to z that are all blocked by distinct vertices in $U' \setminus \{u\}$. It follows that

$$m(U') \geq m^L(U') = L + 2\big|U'\{\succeq L\}\big| = L + 2\big|U'\big| \geq L + 2 \cdot (h + 1 - L) > h + 2 \ , \tag{4.7}$$

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 [Kla85] proves that pyramids possess the Limited hiding-cardinality property, we have the following theorem (with a slight improvement in the constants yielding almost exact bounds).

Theorem 4.20 ([Kla85]). *The black-white pebbling price of the pyramid Π_h of height h is $\mathsf{BW\text{-}Peb}(\Pi_h) = h/2 + \mathrm{O}(1)$. More precisely, for odd-height pyramids we have $\mathsf{BW\text{-}Peb}^{\bullet}(\Pi_{2h+1}) = h + 3$, and for even height we have $\mathsf{BW\text{-}Peb}^{\emptyset}(\Pi_{2h}) = h + 2$.*

Proof. The upper bound on the pebbling price was shown in Lemma 4.7. For the general lower bound, Lemma 4.19 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 4.18, $\mathrm{pot}(\{z\}, \emptyset) \leq 2 \cdot \mathsf{space}(\mathcal{P})$. Thus, $\mathsf{BW\text{-}Peb}^{\bullet}(\Pi_h) \geq \lceil h/2 \rceil + 1$, and it follows that $\mathsf{BW\text{-}Peb}^{\emptyset}(\Pi_h) \geq \lceil h/2 \rceil$.

To get the exact bounds, we have to work a little bit harder. The key observation is that the lower bound obtained from the potential is in fact off by one. Note that the only time the potential can increase is when a black pebble is placed on a source or when a white pebble is removed from a source. If the potential does increase at such a time, we get an upper bound in terms of the space of the involved pebble configurations. We claim that such configurations can never be the ones reaching the maximal space in the pebbling.

To prove this claim, we first make a small local modification of the way $\mathcal{P}$ pebbles sources. By Lemma 3.13, we can assume without loss of generality that $\mathcal{P}$ is frugal. In particular, this means that every pebble placement on a source vertex s at some time t is made either to place a black pebble on a successor or to remove a white pebble from a successor (or both) at some time $t' > t$. Let us change $\mathcal{P}$ so that every pebble placement on a source s is made at time $t' - 1$ immediately before this pebble is needed, and that it is then immediately removed again at time $t' + 1$ (this might lead to sources being pebbled and unpebbled multiple times, but that is not a problem). In this way, every pebble placement on a source s is associated with a unique pebble placement or removal on some $v \in \mathit{succ}(s)$. Let us furthermore stipulate that when the pebble on s is needed for a black pebble placement, then the pebble placed on s is also black, and that it is white when used for a white pebble removal.

Suppose now that the potential increases at time $t' - 1$ due to a black pebble placement on a source. Then the next move is also a pebble placement, and we have $\mathrm{pot}(\mathbb{P}_{t'}) = \mathrm{pot}(\mathbb{P}_{t'-1}) \leq 2 \cdot \mathsf{space}(\mathbb{P}_{t'-1}) = 2 \cdot (\mathsf{space}(\mathbb{P}_{t'}) - 1)$. Similarly, if the potential increases at time $t' + 1$ because of a white pebble removal, the move at time t' is a removal as well, and hence $\mathrm{pot}(\mathbb{P}_{t'+1}) \leq 2 \cdot (\mathit{space}(\mathbb{P}_{t'+1})+1) = 2 \cdot \mathit{space}(\mathbb{P}_{t'}) = 2 \cdot (\mathit{space}(\mathbb{P}_{t'-1}) - 1)$. We see that in both cases the potential underestimates the pebbling space.

Therefore, we can add 1 to the lower bound above, obtaining that $\textit{BW-Peb}^{\bullet}(\Pi_h) \geq \lceil h/2 \rceil + 2 = \lfloor (h-1)/2 \rfloor + 3$, and for odd heights $2h+1$ this bound coincides with the upper bound for binary trees (and hence also pyramids) in Theorem 4.1 yielding the equality $\textit{BW-Peb}^{\bullet}(\Pi_{2h+1}) = h+3$. This in turn implies that $\textit{BW-Peb}^{\emptyset}(\Pi_{2h}) \geq h+2$, which again coincides with the upper bound in Theorem 4.1. For if it would be the case that $\textit{BW-Peb}^{\emptyset}(\Pi_{2h}) \leq h+1$, we could use such a pebbling on the subpyramid rooted at the left predecessor of the sink to get a black pebble on this vertex without exceeding space $h+2$. After this, we would perform a visiting pebbling of the subpyramid rooted at the right predecessor of the sink, and at some suitable time given by Proposition 3.9 move the black pebble up to the sink. This would contradict the bound on the persistently pebbling price for Π_{2h+1} just established. The theorem follows. $\square$

Intriguingly, for even-height pyramids there is still a gap of one between the upper and lower bounds on persistently pebbling price, and similarly for the visiting pebbling price of odd-height pyramids. Although it does not appear to be a terribly important problem, closing this gap would be nice.

Open Problem 1. *Determine exactly* $\textit{BW-Peb}^{\bullet}(\Pi_{2h})$ *and* $\textit{BW-Peb}^{\emptyset}(\Pi_{2h+1})$.

4.5.1 Proving the Limited Hiding-Cardinality Property

We now fill in the the missing link in the proof of Theorem 4.20, i.e., that pyramid graphs possess the Limited hiding-cardinality property. We present the proof 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 4.26 and Lemma 4.30.

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 4.31, but note that the argument does not quite work. However, if we assume a slightly stronger property locally for our disjoint components (Property 4.33), the proof goes through.

3. We then prove this stronger local property by assuming that pyramid graphs have a certain *spreading* property (Definition 4.40 and Theorem 4.41), and by showing in Lemmas 4.39 and 4.42 that the stronger local property holds for such spreading graphs.

4. Finally, in Section 4.5.2, we give a simplified proof of the theorem in [Kla85] 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 4.21 (Tight vertex set). The vertex set U is *tight* if for all $u \in U$ it holds that $u \notin \lceil\!\lceil U \setminus \{u\} \rceil\!\rceil$.

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

Definition 4.22 (Necessary hiding subset). If $x \in \lceil\!\lceil U \rceil\!\rceil$, we define $U\lfloor\!\lfloor x \rfloor\!\rfloor$ to be the subset of U such that for each $u \in U\lfloor\!\lfloor x \rfloor\!\rfloor$ 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 4.22 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., $x \in \lceil\!\lceil U\llbracket x \rrbracket \rceil\!\rceil$.

Lemma 4.23 (Lemma 3.1 in [Kla85]). *If U is tight and $x \in \lceil\!\lceil U \rceil\!\rceil$, then $U\llbracket x \rrbracket$ hides x and this set is also contained in every subset of U that hides x.*

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

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

Lemma 4.24. *For any vertex set U in a layered graph G there is a uniquely determined minimal subset $U^* \subseteq U$ such that $\lceil\!\lceil U^* \rceil\!\rceil = \lceil\!\lceil U \rceil\!\rceil$, U^* is tight, and for any $U' \subseteq U$ with $\lceil\!\lceil U' \rceil\!\rceil = \lceil\!\lceil U \rceil\!\rceil$ it holds that $U^* \subseteq U'$.*

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

Let $L = \mathrm{minlevel}(U)$. 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 $U_L^r = U \setminus \lceil\!\lceil U_L^* \rceil\!\rceil$. Note that there are no vertices on or below level L left in U_L^r, i.e., $U_L^r = U_L^r\{\succ L\}$, and that U_L^* hides the same vertices as does $U\{\preceq L\}$ (since the two sets are equal).

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

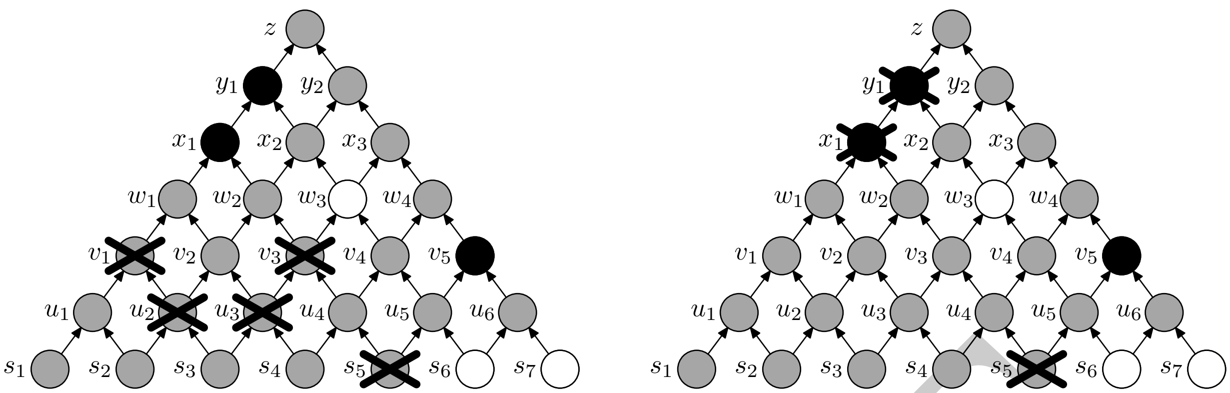
To conclude, we set $U^* = U_M^*$ for $M = \mathrm{maxlevel}(U)$. By construction, the invariant

$$\lceil\!\lceil U_i^* \rceil\!\rceil = \lceil\!\lceil U\{\preceq i\} \rceil\!\rceil \tag{4.8}$$

holds for all levels i. Thus, $\lceil\!\lceil U^* \rceil\!\rceil = \lceil\!\lceil U \rceil\!\rceil$. Also, U^* must be tight since if $v \in U^*$ and $\mathrm{level}(v) = i$, 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 $U^* \nsubseteq U'$. Consider $v \in U^* \setminus U'$ and suppose $\mathrm{level}(v) = i$. On the one hand, we have $v \notin \lceil\!\lceil U_{i-1}^* \rceil\!\rceil$ by construction. On the other hand, by assumption it holds that $v \in \lceil\!\lceil U'\{\prec i\} \rceil\!\rceil$ and thus $v \in \lceil\!\lceil U\{\prec i\} \rceil\!\rceil$. But then by the invariant (4.8) we know that $v \in \lceil\!\lceil U_{i-1}^* \rceil\!\rceil$, 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 4.24 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.



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

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

Figure 7: Illustration of Example 4.25 (with vertices in hiding sets cross-marked).

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 4.25. 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 7(a). Then $U \cup W$ hides B, but U seems unnecessarily large. To get a better hiding set U^*, we can leave s_5 responsible for hiding v_5 but replace $\{v_1, u_2, u_3, v_3\}$ by $\{x_1, y_1\}$. The resulting set $U^* = \{x_1, y_1, s_5\}$ in Figure 7(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 into two disjoint 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 4.26 (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:

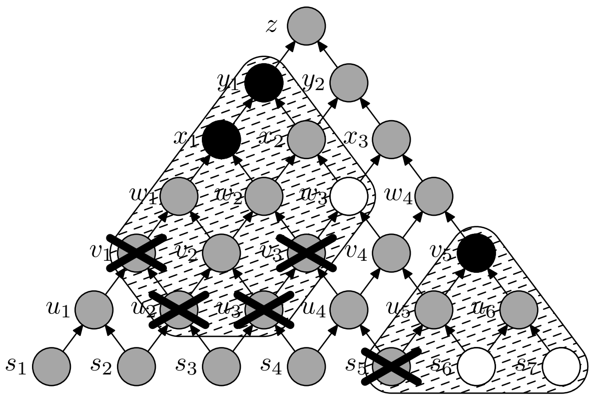
- The set of vertices of $\mathcal{H}$ is $V(\mathcal{H}) = \llbracket X \rrbracket$.

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

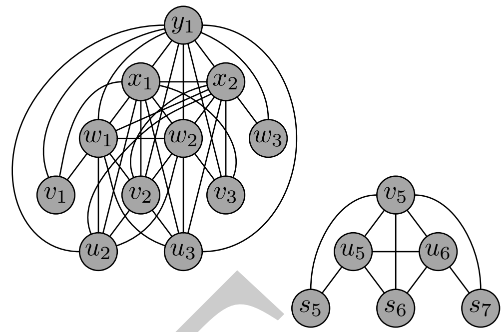
We say that the vertex set X is *hiding-connected* if $\mathcal{H}(G, X)$ is a connected graph.

When the graph G and vertex set X are clear from context, we will sometimes write only $\mathcal{H}(X)$ or even just $\mathcal{H}$. To illustrate Definition 4.26, we give an example.

Example 4.27. Consider again the configuration $(B, W) = (\{x_1, y_1, v_5\}, \{w_3, s_6, s_7\})$ from Example 4.25 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 8(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 8(b). In accordance with the intuition sketched in Example 4.25, $\mathcal{H}(X)$ consists of two connected components.



(a) Vertices hidden by $U \cup W$.



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

Figure 8: Pebble configuration with hiding set and corresponding hiding set graph.

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 4.28). Perhaps a more interesting edge in $\mathcal{H}(X)$ is, for instance, (w_1, x_2). This edge exists since $X\llbracket w_1 \rrbracket = \{v_1, u_2, u_3\}$ and $X\llbracket x_2 \rrbracket = \{u_2, u_3, v_3, w_3\}$ intersect and since as a consequence of this (which is easily verified) we have $\Pi_{\triangle}^{w_1} \cap \lceil\!\lceil X\llbracket w_1 \rrbracket \rceil\!\rceil \cap \Pi_{\triangle}^{x_2} \cap \lceil\!\lceil X\llbracket x_2 \rrbracket \rceil\!\rceil \neq \emptyset$. For the same reason, there is an edge (u_5, u_6) since $X\llbracket u_5 \rrbracket = \{s_5, s_6\}$ and $X\llbracket u_6 \rrbracket = \{s_6, s_7\}$ intersect.

Lemma 4.28. *Suppose for a tight vertex set X that $x \in \lceil\!\lceil X \rceil\!\rceil$ 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 $x, y \in \lceil\!\lceil X \rceil\!\rceil$ by assumption, so x and y are both vertices in $\mathcal{H}(X)$. Since x is above y we have $G_{\triangle}^{x} \supseteq G_{\triangle}^{y}$ and we get $G_{\triangle}^{x} \cap \lceil\!\lceil X\llbracket x \rrbracket \rceil\!\rceil \cap G_{\triangle}^{y} \cap \lceil\!\lceil X\llbracket y \rrbracket \rceil\!\rceil = \lceil\!\lceil X\llbracket x \rrbracket \rceil\!\rceil \cap G_{\triangle}^{y} \cap \{y\} = \{y\} \neq \emptyset$. Thus, (x, y) is an edge in $\mathcal{H}(X)$, so x and y are certainly in the same connected component. □

Corollary 4.29. *If the vertex set X is tight and $x \in \lceil\!\lceil X \rceil\!\rceil$, 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 $V = \lceil\!\lceil X \rceil\!\rceil$, 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 4.30 (Lemma 3.3 in [Kla85]). *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 $V_i = \lceil\!\lceil X \cap V_i \rceil\!\rceil$.*

Proof. We need to show that $V_i = \lceil\!\lceil X \cap V_i \rceil\!\rceil$ 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 $y \in \lceil\!\lceil X \cap V_i \rceil\!\rceil$.

($\Rightarrow$) Suppose $y \in V_i$. Then $X\llbracket y \rrbracket \subseteq V_i$ by Corollary 4.29. Also, $X\llbracket y \rrbracket \subseteq X$ by definition, so $X\llbracket y \rrbracket \subseteq X \cap V_i$. Since $y \in \lceil\!\lceil X\llbracket y \rrbracket \rceil\!\rceil$ by Lemma 4.23, clearly $y \in \lceil\!\lceil X \cap V_i \rceil\!\rceil$.

($\Leftarrow$) Suppose $y \in \lceil\!\lceil X \cap V_i \rceil\!\rceil$. Since X is tight, its subset $X \cap V_i$ must be tight as well. Applying Lemma 4.23 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 4.23 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

$$X\llbracket y \rrbracket = (X \cap V_i)\llbracket y \rrbracket \ . \tag{4.9}$$

By Corollary 4.29, 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 (4.9) into Definition 4.26, 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 pyramids have the LHC property.

Theorem 4.31 (Analogue of Theorem 3.7 in [Kla85]). *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, $\mathrm{pot}_\Pi(\mathbb{P}) = m_\Pi(U)$ 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 4.31. 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 (by Lemma 4.24) 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 4.30 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 4.17 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 4.15 is ill-behaved with respect to unions. Klawe provides the following example of what can happen.

Example 4.32. With vertex labels as in Figures 5 and 6–8, 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 4.33 (Local limited hiding-cardinality property). We say that the pebble configuration $\mathbb{P} = (B, W)$ has the *Local limited hiding-cardinality property*, or just the *Local LHC property* for short, if for any vertex set U such that $U \cup W$ hides B and is hiding-connected, we can find a vertex set U^* such that

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

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

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

Lemma 4.34. *If G has the Local limited hiding-cardinality property 4.33, then G has the Limited hiding-cardinality property 4.17.*

Proof. Consider the tentative proof of Theorem 4.31 and look at the point where it breaks down. If we instead use the Local LHC property to find U_i^*, this time we get that $m\bigl(U_i^*\bigr) \leq m\bigl(U_i\bigr)$ does indeed imply $m\bigl((U \setminus U_i) \cup U_i^*\bigr) \leq m\bigl((U \setminus U_i) \cup U_i\bigr)$, and the theorem follows. □

An obvious way to get the inequality $m(Y \cup U^*) \leq m(Y \cup U)$ in Property 4.33 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 4.15.

Definition 4.35. 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 4.35 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. Ordinary 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 does $m(U) \leq m(V)$ hold globally, but we can also explain locally at each level, by "looking downwards", why U has smaller measure than V.

In Example 4.32, $X_1 \not\precsim_m X_2$ 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 4.32 can no longer occur.

Lemma 4.36 (Lemma 3.4 in [Kla85]). *If $U \precsim_m V$ and $Y \cap V = \emptyset$, then $m(Y \cup U) \leq m(Y \cup V)$.*

Proof. To show that $m(Y \cup U) \leq m(Y \cup V)$, we want to find for each level $j \leq \mathrm{maxlevel}(Y \cup U)$ in U another level i in V 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

$$\begin{aligned} m^j(Y \cup U) = j + 2 \cdot |(U \cup Y)\{\succeq j\}| \leq j + 2 \cdot |U\{\succeq j\}| + 2 \cdot |Y\{\succeq j\}| \leq \\ i + 2 \cdot |V\{\succeq i\}| + 2 \cdot |Y\{\succeq i\}| = m^i(Y \cup V) \end{aligned} \tag{4.10}$$

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 4.35, 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 4.37. 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 4.38. *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 4.37.

Lemma 4.39 (Lemma 3.5 in [Kla85]). *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

$$\left|V^*\right| = |V\{\prec k\}| + |V'| < |V\{\prec k\}| + |V\{\succeq k\}| = |V| \ . \tag{4.11}$$

Combining (4.11) with part 1 of Observation 4.38, 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 (4.11) 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 $\operatorname{maxlevel}(X) - \operatorname{minlevel}(X)$ 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 4.40 (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 $j = 1, \ldots, \operatorname{maxlevel}([\![X]\!])$, the *spreading inequality*

$$|X| \geq L_{\succeq j}([\![X]\!]) + j - \operatorname{minlevel}(X) \tag{4.12}$$

holds.

Let us try to give some more intuition for Definition 4.40 by considering two extreme cases for pyramids:

- 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, (4.12) 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 (4.12) 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 4.8 and Observation 4.10.

Very loosely, Definition 4.40 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 (4.12) does indeed hold for pyramids.

Theorem 4.41 ([Kla85]). *Pyramids are spreading graphs.*

Unfortunately, the proof of Theorem 4.41 in [Kla85] 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 4.5.2, we give a simplified, direct proof of the fact that pyramids are spreading that might be of independent interest.

Accepting Theorem 4.41 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 4.33 to hold, then replacing U by the minimum-size hiding set in Lemma 4.39 we get a hiding set in accordance with the Local LHC property.

Lemma 4.42 (Lemma 3.6 in [Kla85]). *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 4.42 for a moment, let us note that if we combine this lemma with Lemma 4.36 and Theorem 4.41, the Local limited hiding-cardinality property for pyramids follows.

Corollary 4.43. *Pyramid graphs have the Local limited hiding-cardinality property 4.33.*

Proof of Corollary 4.43. This is more or less immediate, but we write down the details for completeness. Since pyramids are spreading by Theorem 4.41, Lemma 4.42 says that U^* is a hiding set for (B, W) and that $U^* \precsim_m U$. Lemma 4.36 then yields that $m(Y \cup U^*) \leq m(Y \cup U)$ for all Y with $Y \cap U = \emptyset$. Finally, Lemma 4.42 also tells us that $U^* \subseteq B$ or $|U^*| < |B| + |W|$, and thus all conditions in Property 4.33 are satisfied. □

Continuing by plugging Corollary 4.43 into Lemma 4.34, we get the global LHC property in Theorem 4.31 on page 28. 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 4.41 and Lemma 4.42. We attend to Lemma 4.42 right away, deferring a proof of Theorem 4.41 to Section 4.5.2.

Proof of Lemma 4.42. 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 4.39. 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 $m^j\big(U^*\big) \leq m^i(U)$. Perhaps it will be helpful at this point for the reader to look at Example 4.25 again, where the replacement of $U_1 = \{v_1, u_2, u_3, v_3\}$ in Figure 7(a) by $U_1^* = \{x_1, y_1\}$ in Figure 7(b) shows Lemmas 4.39 and 4.42 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 4.25). 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 $m^j\big(U^*\big)$, 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 $m^L\big(U\big)$.

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

$$
\begin{aligned}
m^j\big(U^*\big) &= j + 2 \cdot \big|U^*\{\succeq j\}\big| && \big[\text{ by definition of } m^j(\cdot) \,\big] \\
&\leq j + 2 \cdot \big|U^*\big| && \big[\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V \,\big] \\
&\leq j + 2 \cdot |B| && \big[\text{ by construction of } U^* \text{ in Lemma 4.39} \,\big] \\
&\leq j + 2 \cdot |U| && \big[\text{ by assumption } |U| \geq |B| + |W| \geq |B| \,\big] \\
&= j + 2 \cdot \big|U\{\succeq j\}\big| && \big[\, U\{\succeq j\} = U \text{ since } j \leq \mathrm{minlevel}(U) \,\big] \\
&= m^j(U) && \big[\text{ by definition of } m^j(\cdot) \,\big]
\end{aligned}
$$

and we can choose $i = j$ in Definition 4.35.

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

$$
L_{\succeq j}(B) \leq L_{\succeq j}\big(\llbracket U \cup W \rrbracket\big) \tag{4.13}
$$

holds by part 2 of Observation 4.38. Moreover, $U \cup W$ is a hiding-connected set of vertices in a spreading graph G, so the spreading inequality (4.12) says that $|U \cup W| \geq L_{\succeq j}\big(\llbracket U \cup W \rrbracket\big) + j - L$, or

$$
j + L_{\succeq j}\big(\llbracket U \cup W \rrbracket\big) \leq L + |U \cup W| \tag{4.14}
$$

after reordering. Combining (4.13) and (4.14) we have that

$$
j + L_{\succeq j}(B) \leq L + |U \cup W| \tag{4.15}
$$

and it follows that

$$\begin{aligned}
m^j(U^*) &= j + 2 \cdot \left|U^*\{\succeq j\}\right| && \left[\text{ by definition of } m^j(\cdot)\ \right] \\
&\le j + \left|U^*\{\succeq j\}\right| + \left|U^*\right| && \left[\text{ since } V\{\succeq j\} \subseteq V \text{ for any } V\ \right] \\
&\le j + L_{\succeq j}(B) + |B| && \left[\text{ by construction of } U^* \text{ in Lemma 4.39}\ \right] \\
&\le L + |U \cup W| + |B| && \left[\text{ by the inequality (4.15)}\ \right] \\
&\le L + 2 \cdot |U| && \left[\text{ by assumption } |U| \ge |B| + |W|\ \right] \\
&= L + 2 \cdot |U\{\succeq L\}| && \left[\ U\{\succeq L\} = U \text{ since } L \le \mathrm{minlevel}(U)\ \right] \\
&= m^L(U) && \left[\text{ by definition of } m^L(\cdot)\ \right]
\end{aligned}$$

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 4.35.

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

Concluding this part of the proof construction, we want to make a comment about Lemmas 4.39 and 4.42 and try to rephrase what they say about hiding sets. Given a tight set $U \cup W$ such that $B \subseteq [\![U \cup W]\!]$, we can always pick a U^* as in Lemma 4.39 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 4.42, for $j > \mathrm{minlevel}(U \cup W)$ we can always prove that

$$m^j(U^*) \le \mathrm{minlevel}(U \cup W) + |U| + (|B| + |W|) \ . \tag{4.16}$$

The key message of Lemma 4.42 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 $L = \mathrm{minlevel}(U \cup W)$.

4.5.2 Proving That Pyramids Are Spreading Graphs

The fact that pyramids are spreading graphs, that is, that they satisfy the inequality (4.12), is a consequence of the following lemma.

Lemma 4.44 (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 $V = [\![X]\!]$, then there is a unique vertex $x \in V$ such that $X = X\llbracket x\rrbracket$ and $V = [\![X\llbracket x\rrbracket]\!] \subseteq \Pi^x_\Delta$.*

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 8 are both examples of such "ice-cream cones."

Before proving Lemma 4.44, we show how this lemma can be used to establish that pyramid graphs are spreading by a converging-paths argument as in Observation 4.10.

Proof of Theorem 4.41. 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 $V = [\![X]\!]$. Let $x \in V$ be the vertex given by Lemma 4.44 such that $X = X\llbracket x\rrbracket$ and $V = [\![X\llbracket x\rrbracket]\!] \subseteq \Pi^x_\Delta$, and let $M = \mathrm{level}(x)$.

For any $j \le M$ we have

$$L_{\succeq j}([\![X]\!]) \le M - j + 1 \ . \tag{4.17}$$

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 $[\![X]\!]$ above level j since $[\![X]\!] \subseteq \Pi^x_\Delta$.

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 $L = \mathrm{minlevel}(X)$. 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 4.10. 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

$$|X| \geq M - L + 1 \tag{4.18}$$

follows. By combining (4.17) and (4.18), we get that

$$|X| - L_{\succeq j}(\lceil\!\lceil X \rceil\!\rceil) \geq M - L + 1 - (M - j + 1) = j - L \tag{4.19}$$

which is the required spreading inequality (4.12). □

The rest of this subsubsection 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 4.45. *Suppose for a planar DAG G that we have a source path P to a vertex w and two vertices $u, v \in G^{\rightsquigarrow w}_{\vartriangle}$ 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 5 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 4.46. *Suppose that X is a tight set of vertices in a pyramid Π and that $v \in \lceil\!\lceil X \rceil\!\rceil$. Then $\lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil \subseteq \Pi^v_{\vartriangle}$.*

Proof. Since all vertices in $X\llbracket v \rrbracket$ have a path to v by definition, it holds that $X\llbracket v \rrbracket \subseteq \Pi^v_{\vartriangle}$. Any vertex $u \in \Pi \setminus \Pi^v_{\vartriangle}$ 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, $\left(\Pi \setminus \Pi^v_{\vartriangle}\right) \cap \lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil = \emptyset$. □

Observation 4.47. *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 4.46 and 4.47, we can simplify the definition of the hiding set graph. Note that Observation 4.46 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 4.48. *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., $\Pi^u_{\vartriangle} \cap \lceil\!\lceil X\llbracket u \rrbracket \rceil\!\rceil \cap \Pi^v_{\vartriangle} \cap \lceil\!\lceil X\llbracket v \rrbracket \rceil\!\rceil \neq \emptyset$.*

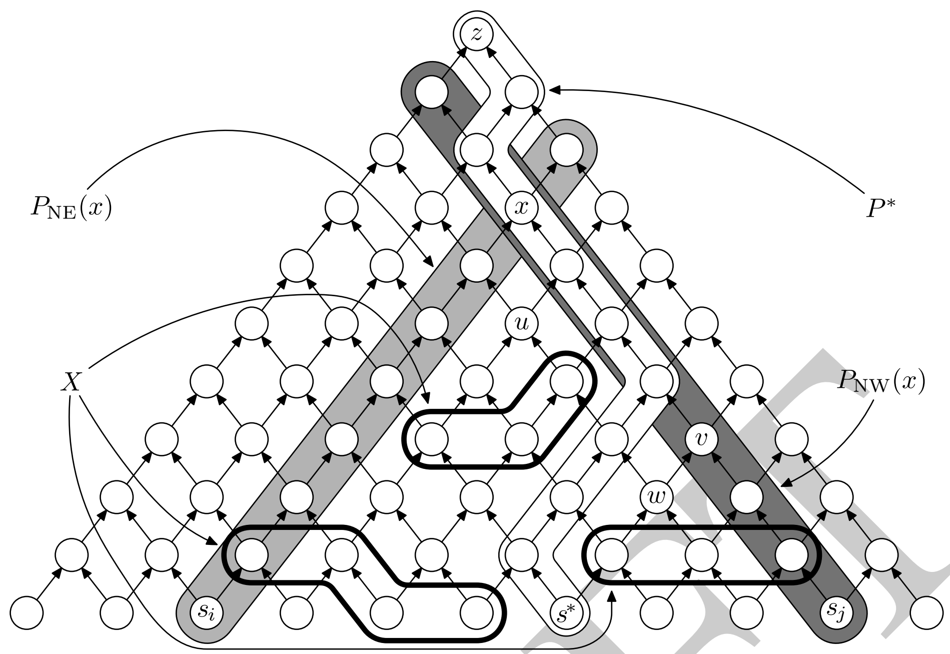


Figure 9: Illustration of proof of Lemma 4.49 that $\mathcal{H}$ is not connected if $x \notin \lceil\!\lceil X \rceil\!\rceil$.

2. $\lceil\!\lceil X\lfloor\!\lfloor u \rfloor\!\rfloor \rceil\!\rceil \cap \lceil\!\lceil X\lfloor\!\lfloor v \rfloor\!\rfloor \rceil\!\rceil \neq \emptyset$.

3. $X\lfloor\!\lfloor u \rfloor\!\rfloor \cap X\lfloor\!\lfloor v \rfloor\!\rfloor \neq \emptyset$.

Proof. The directions (1) $\Rightarrow$ (2) and (3) $\Rightarrow$ (2) are immediate. The implication (2) $\Rightarrow$ (1) also follows easily, since $\lceil\!\lceil X\lfloor\!\lfloor u \rfloor\!\rfloor \rceil\!\rceil \subseteq \Pi^u_\Delta$ and $\lceil\!\lceil X\lfloor\!\lfloor v \rfloor\!\rfloor \rceil\!\rceil \subseteq \Pi^v_\Delta$ by Observation 4.46. To prove (2) $\Rightarrow$ (3), fix some vertex $w \in \lceil\!\lceil X\lfloor\!\lfloor u \rfloor\!\rfloor \rceil\!\rceil \cap \lceil\!\lceil X\lfloor\!\lfloor v \rfloor\!\rfloor \rceil\!\rceil$ and let P be a source path to w as in Observation 4.47 with $P \cap X = \{y\}$ for some vertex y. Since $P \cap X\lfloor\!\lfloor u \rfloor\!\rfloor \neq \emptyset \neq P \cap X\lfloor\!\lfloor u \rfloor\!\rfloor$ by assumption, we have $y \in X\lfloor\!\lfloor u \rfloor\!\rfloor \cap X\lfloor\!\lfloor v \rfloor\!\rfloor \neq \emptyset$. □

As the first part of the proof of Lemma 4.44, 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 4.49. *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 $\lceil\!\lceil X \rceil\!\rceil \subseteq \Pi^x_\Delta$.*

Proof. It is clear that at most one vertex $x \in \lceil\!\lceil X \rceil\!\rceil$ 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 $\lceil\!\lceil X \rceil\!\rceil \subseteq \Pi^x_\Delta$. 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 $x \notin \lceil\!\lceil X \rceil\!\rceil$, 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 9. 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 $P_{\mathrm{NE}}(s_i) \cap \lceil\!\lceil X \rceil\!\rceil \neq \emptyset$. Similarly, consider $P_{\mathrm{NW}}(s_{h+1}), P_{\mathrm{NW}}(s_h), \ldots$ and let s_j be the first vertex such that $P_{\mathrm{NW}}(s_j) \cap \lceil\!\lceil X \rceil\!\rceil \neq \emptyset$. 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 $\lceil\!\lceil X \rceil\!\rceil \subseteq \Pi^x_\Delta$, since no NE-path to the left of $P_{\mathrm{NE}}(s_i) = P_{\mathrm{NE}}(x)$ intersects $\lceil\!\lceil X \rceil\!\rceil$ 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 $x \in \lceil\!\lceil X \rceil\!\rceil$.

To derive a contradiction, suppose instead that $x \notin \lceil\!\lceil X \rceil\!\rceil$. By definition, there is a path P from some source s^* to x such that $P \cap \lceil\!\lceil X \rceil\!\rceil = \emptyset$. P cannot coincide with $P_{\mathrm{NE}}(x)$ or $P_{\mathrm{NW}}(x)$ since the latter two paths both intersect $\lceil\!\lceil X \rceil\!\rceil$ by construction. Since $\Pi^\nabla_x \cap \lceil\!\lceil X \rceil\!\rceil = \emptyset$, we can extend P to a path $P^* : s^* \rightsquigarrow z$ via x having the property that $P^* \cap \lceil\!\lceil X \rceil\!\rceil = \emptyset$ but there are vertices in $\mathcal{H}(X)$ both to the left and to the right of P^*, namely, the non-empty sets $P_{\mathrm{NE}}(x) \cap \lceil\!\lceil X \rceil\!\rceil \cap \Pi^x_\Delta$ and $P_{\mathrm{NW}}(x) \cap \lceil\!\lceil X \rceil\!\rceil \cap \Pi^x_\Delta$. 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 $x \in \lceil\!\lceil X \rceil\!\rceil$ 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 4.48 says that $\lceil\!\lceil X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \rceil\!\rceil \cap \lceil\!\lceil X_{\lfloor\!\lfloor v \rfloor\!\rfloor} \rceil\!\rceil \neq \emptyset$. Pick any vertex $w \in \lceil\!\lceil X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \rceil\!\rceil \cap \lceil\!\lceil X_{\lfloor\!\lfloor v \rfloor\!\rfloor} \rceil\!\rceil$ 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 9, which illustrate the fact that $w \notin \lceil\!\lceil X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \rceil\!\rceil$ if $w \in \lceil\!\lceil X_{\lfloor\!\lfloor v \rfloor\!\rfloor} \rceil\!\rceil$.

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 $y \in P_{\mathrm{NW}}(w) \cap X_{\lfloor\!\lfloor u \rfloor\!\rfloor} \cap \Pi^w_\Delta$ and note that y must also be located to the right of P^*. By Definition 4.22, 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 4.45.) Consider the source path that starts like P^* and then switches to P' at some intersection point in $P' \cap P^* \cap \Pi^\nabla_y$. 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 4.44 is that all vertices in X are required to hide the top vertex $x \in \lceil\!\lceil X \rceil\!\rceil$ found in Lemma 4.49.

Lemma 4.50. *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 $\lceil\!\lceil X \rceil\!\rceil \subseteq \Pi^x_\Delta$. 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 4.23 says that $X_{\lfloor\!\lfloor u' \rfloor\!\rfloor} \subseteq X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$ and Proposition 4.48 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 10 for an illustration of the proof from here onwards.

By Definition 4.22, 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 $r \in X_{\lfloor\!\lfloor x \rfloor\!\rfloor}$. 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

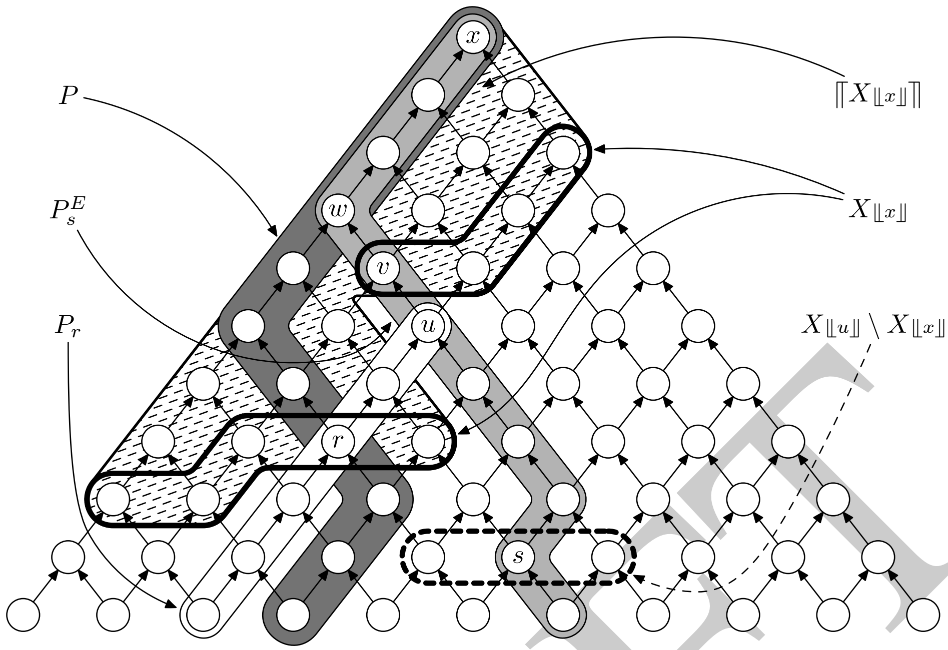


Figure 10: Illustration of proof of Lemma 4.50 that all of X is needed to hide x.

can start with P_s and then switch to P when the paths intersect, which implies that $s \in X\llfloor x \rrfloor$ 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 $s \notin X\llfloor x \rrfloor$. 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 $x \in X\llfloor x \rrfloor$. 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 4.44 now follows. Thereby, the proof of the lower bound on the black-white pebbling price of pyramid graphs in Theorem 4.20 on page 23 is complete. Since the proof of the pyramid graph lower bound is arguably the most involved construction in this whole survey, it is natural to conclude this section by asking whether the construction can be simplified.

Open Problem 2. *Is there a simpler way than in [Kla85] to prove that* $\textsf{BW-Peb}(\Pi_h) = h/2 + \mathrm{O}(1)$*?*

5 Superconcentrators

Superconcentrators are graphs that solve the generic problem of connecting N clients to N servers in a setting where either the clients or the servers are interchangeable (so that it does not matter which client is connected to which server). Superconcentrators are mainly used in network theory, but these graphs have also found numerous applications in theoretical computer science. As we will see in the remainder of this survey, superconcentrators are an important building block in many pebbling results.

Definition 5.1 (Superconcentrator). A directed acyclic graph G is an N-*superconcentrator* if it has N sources $S = \{s_1, \dots, s_N\}$, N sinks $Z = \{z_1, \dots, z_N\}$, and for any subsets S' and Z' of sources and sinks of size $|S'| = |Z'| = m$ it holds that there are m vertex-disjoint paths between S' and Z' in G.

Note that we do not assume that we can specify which source in S' should be connected to which sink in Z'.

5.1 A Simple (Non-optimal) Superconcentrator Construction

Clearly, any complete balanced bipartite graph $K_{N,N}$ is an N-superconcentrator in the sense of Definition 5.1, but the point is that we want to build superconcentrators with as few edges as possible, or as low *density* as possible measured as the number of edges divided by N. That is, we want our superconcentrators to be very sparse graphs that nevertheless have almost as good connectivity properties as complete bipartite graphs. In addition, for our pebbling purposes we will want the superconcentrators to have bounded indegree, but this extra requirement is easy to take care of (as will be seen below).

As a starting point for our discussion, and to give the reader a concrete example, we note that there is a fairly simple recursive construction of an N-superconcentrator with $\Omega(N \log N)$ edges, fan-in 2, and depth $\mathrm{O}(\log N)$. These parameters can be achieved by gluing together two butterfly graphs back to back with the edge directions in the second copy reversed as shown in Figure 11, where for $N = 2^n$ the n-dimensional butterfly graph is a DAG with vertices labelled by pairs (w, i) for $0 \leq w \leq 2^n - 1$ and $0 \leq i \leq n$, and with edges from vertex (w, i) to $(w', i+1)$ if the binary representations of w and w' are equal except for possibly in the $(i + 1)$st most significant bit.

These graphs in fact satisfy the stronger property that it is possible to specify which source should be connected to which sink. Such graphs that can route the sources to any permutation of the sinks are known as *connectors*.

Proposition 5.2. *The graph $H(N)$ for $N = 2^n$ constructed by connecting two n-dimensional butterfly graphs as in Figure 11 is an N-connector.*

Proof. The proof is by induction. The base case is the complete bipartite graph $H(2) = K_{2,2}$ which clearly can route any permutation of $\{0, 1\}$.

As a slightly more challenging example, consider four sources and sinks and suppose that $(w_j, 0)$ is the source in $H(4)$ that should be routed to the jth sink from the top for $0 \leq j \leq 3$. Route $(w_0, 0)$ to the upper copy of $K_{2,2}$ in $H(4)$ and $(w_2, 0)$ to the lower copy. Then there are free vertices and edges so that $(w_1, 0)$ and $(w_3, 0)$ can be routed to oppsite copies of $H(2) = K_{2,2}$. Inside the $H(2)$-copies, use crossing edges if necessary so that the paths for $(w_0, 0)$ and $(w_2, 0)$ end up on even levels and the paths for $(w_1, 0)$ and $(w_3, 0)$ on odd levels (counted from the topmost level 0). Then in the final step, $(w_0, 0)$ and $(w_2, 0)$ have been routed to correct levels and should continue on straight edges, and the positions to which $(w_1, 0)$ and $(w_3, 0)$ have been routed can be interchanged by crossing edges if necessary so that these two paths end up on the correct levels as well.

The general argument for $N = 2^n$ is similar. Again, let $(w_j, 0)$ be the source that should be routed to the jth sink for $0 \leq j \leq 2^n - 1$. Route $(w_0, 0)$ to its successor u in the upper copy of $H(N/2)$ and $(w_{N/2}, 0)$ to its successor v in the lower copy of $H(N/2)$. Then the other predecessor of u must be sent to the lower copy, and the other predecessor of v to the upper copy. However, it can be verified that for each pair of sources $(w_j, 0)$ and $(w_{j+N/2}, 0)$ we can make sure they are sent to opposite copies of $H(N/2)$. By induction, inside these subgraphs all sources can be routed to the correct levels except for possibly the most

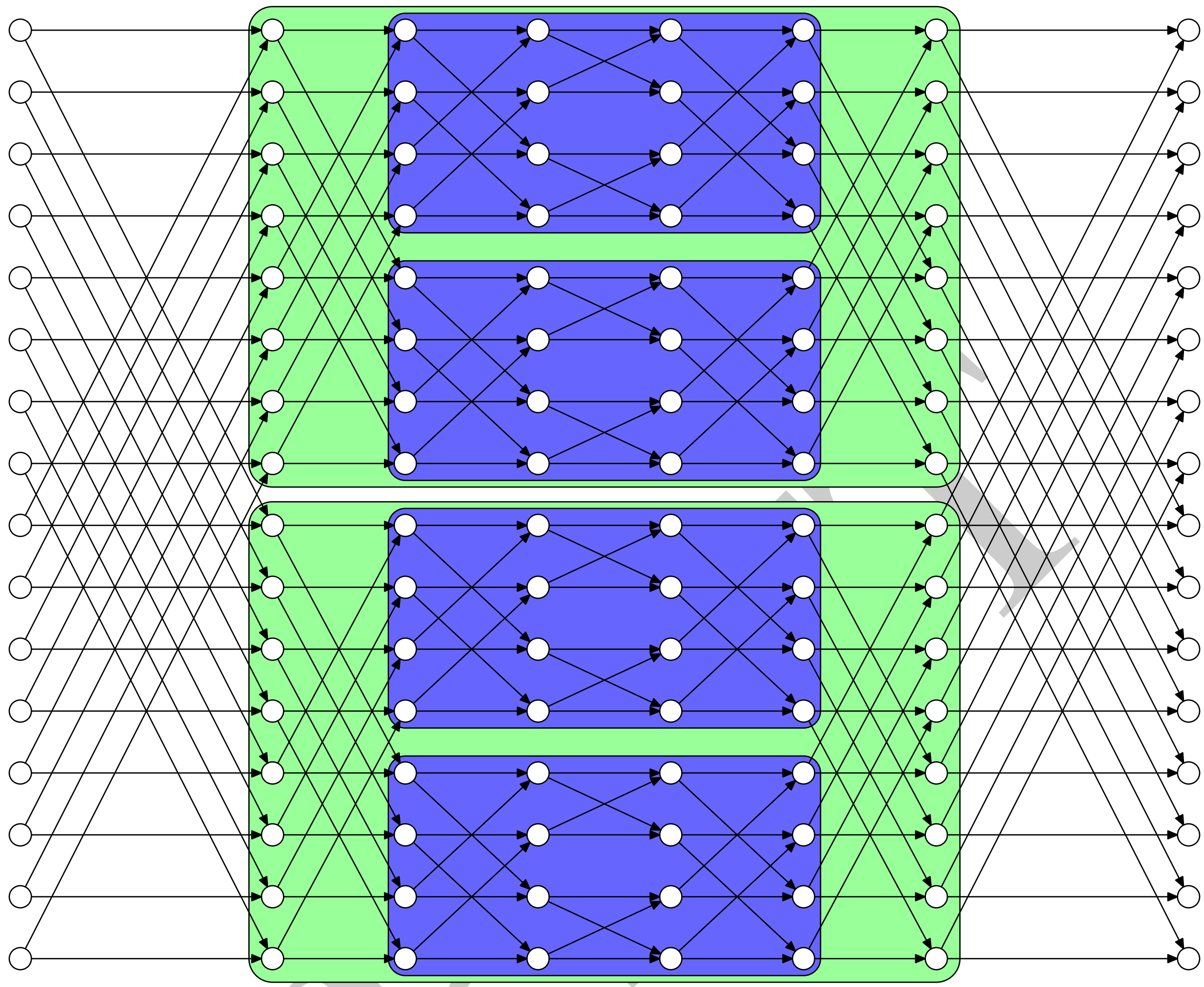


Figure 11: Example of recursive superconcentrator construction $H(16)$ using butterfly graphs.

significant bit in their binary representation. But this bit can be corrected in the final step by letting the paths of $(w_j, 0)$ and $(w_{j+N/2}, 0)$ cross if necessary. □

It is known that connectors require $\Omega(N \log N)$ edges [PV76]. One can ask whether this lower bound applies for superconcentrators as well, or whether the more relaxed connectivity requirements allows us to do better. Valiant [Val76] proved that the latter case holds by establishing the existence of superconcentrators of constant density, i.e., with number of edges linear in N (and hence with a linear number of vertices as well). We will refer to such graphs as *linear superconcentrators*. This result refuted an earlier conjecture that N-superconcentrators should require $\Omega(N \log N)$ edges.

Gabber and Galil [GG81] provided the first explicit construction of linear superconcentrators, based on an earlier non-explicit construction by Pippenger [Pip77]. We remark that the superconcentrators in [GG81] also have logarithmic depth. The currently best known explicit construction (i.e., with lowest density) that we are aware of is a family of superconcentrators of density 44 due to Alon and Capalbo [AC03]. We state the parameters for this construction in Section 5.4. For non-explicit constructions, the current world record holder seems to be Schöning [Sch05] with a density of 28. The best lower bound on which density can be achieved for superconcentrators without degree restrictions still appears to be $5 - o(1)$ as proven by Lev and Valiant [LV83]. In the same paper it is also shown that N-superconcentrators of indegree 2 must have

$(4 - o(1))N$ vertices.

5.2 A Generic Lower Bound on Superconcentrator Trade-offs

Let us next prove a generic pebbling trade-off that holds for any superconcentrator regardless of how it is constructed. The trade-off follows immediately from the next two lemmas, which in a sense explain why superconcentrators are good building blocks if we want to construct graphs that are hard to pebble.

Lemma 5.3 ([PTC77]). *Suppose that the DAG G is an N-superconcentrator, that $\mathbb{P}$ is a pebble configuration on G with at most s pebbles, and that Z' is a set of strictly more than s sinks of G. Then at least $N - s$ sources have completely pebble-free paths to Z'.*

Proof. Any set S' of $|S'| = s + 1$ sources have vertex-disjoint paths to some subset of sinks in Z', and at most s of these paths can be blocked by pebbles in $\mathbb{P}$. Thus, at most s sources can lack completely pebble-free paths to Z'. □

Lemma 5.4 (Basic Lower Bound Argument ([GT78, LT82])). *Suppose that $\mathcal{P} = \{\mathbb{P}_\sigma, \mathbb{P}_{\sigma+1}, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling of an N-superconcentrator such that $\mathsf{space}(\mathbb{P}_\sigma) \leq s_\sigma$, $\mathsf{space}(\mathbb{P}_\tau) \leq s_\tau$, and $\mathcal{P}$ pebbles at least $s_\sigma + s_\tau + 1$ sinks during the closed time interval $[\sigma, \tau]$. Then $\mathcal{P}$ pebbles and unpebbles at least $N - s_\sigma - s_\tau$ different sources during the open time interval (σ, τ).*

Proof. By Lemma 5.3, at least $N - s_\sigma - s_\tau$ sources have paths to the pebbled sinks such that these paths are pebble-free at times σ and τ. It follows from Lemma 3.11 that all of these sources must be pebbled during the time interval (σ, τ). □

Following Lengauer and Tarjan, we will refer to Lemma 5.4 as the *Basic Lower Bound Argument* lemma, or just BLBA-lemma, for superconcentrators. As a corollary of the BLBA-lemma, we get the following lower bound on time-space trade-offs for superconcentrator pebblings.

Theorem 5.5 ([LT82]). *Any complete black-white pebbling of an N-superconcentrator in space at most s has to pebble at least $\Omega(N^2/s)$ sources (so, in particular, this is a lower bound on the pebbling time).*

Proof. Apply Lemma 5.4 $\lfloor N/(2s + 1) \rfloor$ times. □

5.3 A Trade-off Upper Bound for a Linear Superconcentrator Family

There are superconcentrator constructions for which the lower bound argument in Lemma 5.4 and Theorem 5.5 is tight; for instance, this is the case for the butterfly-style connectors Section 5.1 [LT82, SS77]. For linear superconcentrators, however, no such efficient pebbling strategies are known. In what follows, we will focus on the construction by Pippenger [Pip77] and show that such graphs can be pebbled somewhat efficiently. As noted above, Pippenger's construction was later made explicit by Gabber and Galil [GG81], but the parameters in the explicit version are not as good and the pebbling strategy has a corresponding deterioration in performance. What we will get in the end are results of the following type.

Theorem 5.6 ([LT82]). *There are explicitly constructible N-superconcentrators of constant density that can be completely black-pebbled with s pebbles, $\Omega(\log N) = s = \mathrm{O}(N)$, in time $\mathrm{O}\big(N \cdot (N/s)^k\big)$ for some constant $k \in \mathbb{R}^+$.*

However, for the explicit superconcentrators in [GG81] the constant obtained in [LT82] is $k = 156.67$, which leaves quite a gap compared with the lower bound exponent $k = 1$ in Theorem 5.5. Even for the non-explicit superconcentrators in [Pip77] the upper bound is for $k = 9.84$. Lengauer and Tarjan mention the tightening of this gap as an open problem.

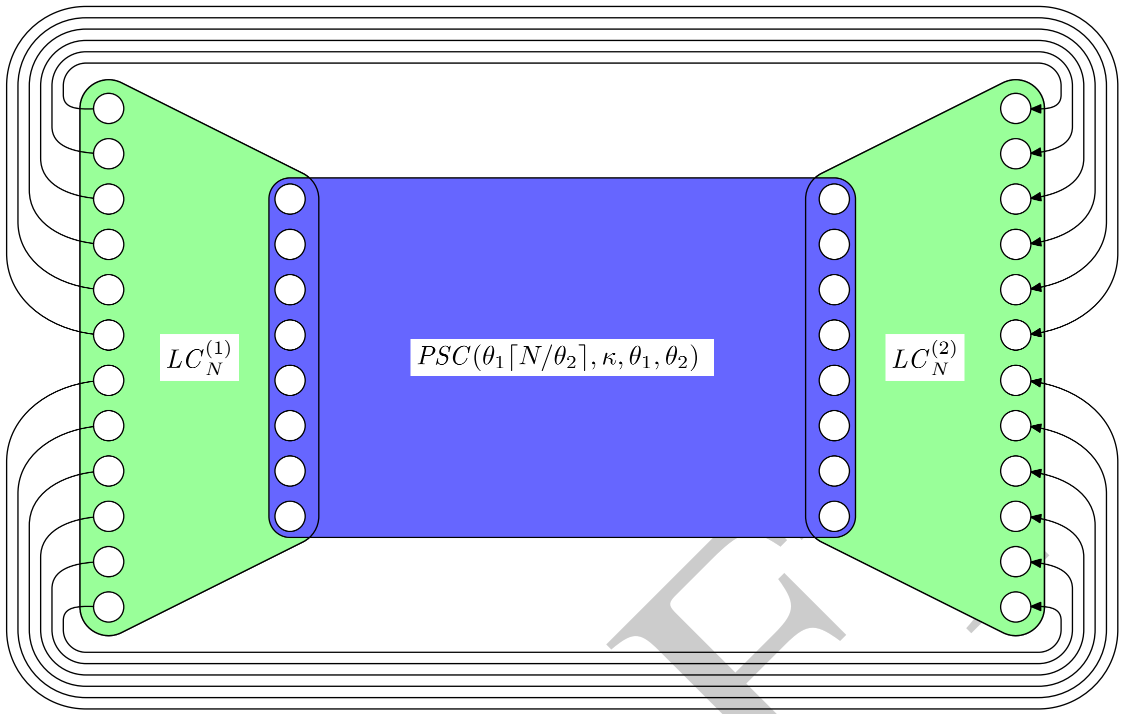


Figure 12: Schematic illustration of Pippenger's superconcentrator $PSC(N, \kappa, \theta_1, \theta_2)$.

Open Problem 3 ([LT82]). *Is it possible to tighten the gap in the exponent between the black-white pebbling lower bound in Theorem 5.5 and the black pebbling upper bound in Theorem 5.6 for some explicit or non-explicit superconcentrator family?*

For instance, one could ask what happens if one tries to apply the proof techniques in [LT82] to more recent constructions such as the ones in [AC03, Sch05].

But let us return to the superconcentrators in [GG81, Pip77]. To describe these constructions, we need a particular kind of bipartite graphs, the concept of which seems to have originated with Pinsker [Pin73]. For a bipartite graph with left vertices U and right vertices V, let us write $N(U')$ to denote the set of neighbours in V of a subset of vertices $U' \subseteq U$ on the left, i.e., the vertices in V which are connected by edges to vertices in U'.

Definition 5.7 (Linear concentrator). Let $n, \kappa, \theta_1, \theta_2$ be positive integers such that $\theta_1 < \theta_2$ and $\theta_1 \mid \theta_2\kappa$, and set $\kappa' = \theta_2\kappa/\theta_1$. Then an $(n, \kappa, \theta_1, \theta_2)$*-linear concentrator* is a bipartite graph with n left vertices, $\theta_1\lceil n/\theta_2\rceil$ right vertices, left vertex degree at most κ, and right vertex degree at most κ', which furthermore has the property that every left vertex set U' of size $\lvert U'\rvert \leq n/2$ has $\lvert N(U')\rvert \geq \lvert U'\rvert$ neighbours.

Obviously, an $(n, \kappa, \theta_1, \theta_2)$-linear concentrator has at most κn edges. Also, it follows from the requirements in the definition that $\theta_1 \gtrapprox \theta_2/2$ (since otherwise left vertex sets of size $n/2$ could not have $n/2$ neighbours on the right). Pippenger uses these graphs to construct superconcentrators as described next (see Figure 12 for an illustration).

Definition 5.8 (Pippenger's superconcentrator [Pip77]). Let $PSC(N, \kappa, \theta_1, \theta_2)$ be the following recursively defined graph:

1. If $N \leq \theta_1\theta_2$, then $PSC(N, \kappa, \theta_1, \theta_2)$ is the complete bipartite graph $K_{N,N}$ with edges directed from left to right.

2. Otherwise, $PSC(N, \kappa, \theta_1, \theta_2)$ is a graph with N sources $s_1, \ldots, s_N$ and N sinks $z_1, \ldots, z_N$ such that:
   (a) There are direct edges (s_i, z_i), $i = 1, \ldots, N$, from the sources to the sinks.
   (b) The sources are the left vertices of an $(N, \kappa, \theta_1, \theta_2)$-linear concentrator $LC_N^{(1)}$ with edges directed from left to right.
   (c) The sinks are also left vertices of an $(N, \kappa, \theta_1, \theta_2)$-linear concentrator $LC_N^{(2)}$, but mirrored so that the edges are going in the other direction.
   (d) The right vertices of $LC_N^{(1)}$ and (its mirror image) $LC_N^{(2)}$ are connected via a copy of the graph $PSC(\theta_1\lceil N/\theta_2\rceil, \kappa, \theta_1, \theta_2)$, where we identify the right vertices of $LC_N^{(1)}$ with the sources of $PSC(\theta_1\lceil N/\theta_2\rceil, \kappa, \theta_1, \theta_2)$, and the right vertices of $LC_N^{(2)}$ with the sinks.

Since we will have to expand the recursive construction multiple steps in the proofs that will follow, for ease of notation let us define $\lambda(N) = \lambda^1(N) = \theta_1\lceil N/\theta_2\rceil$ to be the number of sources and sinks in the subgraph superconcentrator in Definition 5.8, and for the jth level of recursion define $\lambda^j(N) = \theta_1\lceil \lambda^{j-1}(N)/\theta_2\rceil$. Let us verify that $PSC(N, \kappa, \theta_1, \theta_2)$ is indeed a superconcentrator and write down some of its properties.

Lemma 5.9. *The graph $PSC(N, \kappa, \theta_1, \theta_2)$ is an N-superconcentrator with $\frac{2}{1-\theta_1/\theta_2}N + \mathrm{O}(\log N)$ vertices, $\frac{2\kappa+1}{1-\theta_1/\theta_2}N + \mathrm{O}(\log N)$ edges, bounded indegree, and depth $\mathrm{O}(\log N)$.*

Proof sketch. That $PSC(N, \kappa, \theta_1, \theta_2)$ has bounded indegree is immediate from the construction. To see that it is a superconcentrator, let S' and Z' be any sets of sources and sinks of $PSC(N, \kappa, \theta_1, \theta_2)$ with $|S'| = |Z'| = m$. If $m > N/2$ then clearly we can route at least $m - N/2$ pairs of sources and sinks via direct edges, Let us therefore assume that $m \leq N/2$ and see how S' can be routed to Z' without using any direct edges. By combining the definition of linear concentrators with Hall's theorem, we can conclude that there is a perfect matching between the vertices in S' and some vertex set $S'' \subseteq N(S')$ of size m in $LC_N^{(1)}$ and also between Z' and some set $Z'' \subseteq N(Z')$ in $LC_N^{(2)}$. By induction, we can route S'' to Z'' in $PSC(\lambda(N), \kappa, \theta_1, \theta_2)$ since this graph is a superconcentrator. Hence, $PSC(N, \kappa, \theta_1, \theta_2)$ is a superconcentrator as well.

To establish the bounds on the number of vertices and edges in $PSC(N, \kappa, \theta_1, \theta_2)$, we need to calculate how many recursive steps are needed to reach the base case in Definition 5.8. Observe that if the size of the subgraph superconcentrator in the construction were $\theta_1 N/\theta_2$ rather than $\theta_1\lceil N/\theta_2\rceil$, then the subgraphs at the jth level of recursion would have size roughly $(\theta_1/\theta_2)^j N$ and after a logarithmic number of steps we would be down to constant size. In this case, the bounds as stated in the lemma would hold without the additive $\mathrm{O}(\log N)$ adjustment.

This turns out to be a helpful bit of intuition. In fact, one can prove that $\lambda^j(N)$ behaves almost like $(\theta_1/\theta_2)^j N$ except for at most a constant term. Therefore, the number of recursive levels will indeed be logarithmic, and at each level we are only off by at most an additive constant. Hence, the bounds hold if we adjust by a term $\mathrm{O}(\log N)$. To prove this formally, a useful inequality is

$$\left(\frac{\theta_1}{\theta_2}\right)^j N \leq \lambda^j(N) \leq \left(\frac{\theta_1}{\theta_2}\right)^j N + \frac{\theta_1(\theta_2 - 1)}{\theta_2 - \theta_1} \tag{5.1}$$

which can be established by an inductive argument based on the fact that $\lceil N/\theta_2\rceil \leq N/\theta_2 + (\theta_2 - 1)/\theta_2$. Using that $\frac{\theta_1(\theta_2-1)}{\theta_2-\theta_1} < \frac{\theta_1\theta_2}{\theta_2-\theta_1} \leq \theta_1\theta_2$, we can rewrite (5.1) in the slightly weaker form

$$\left(\frac{\theta_1}{\theta_2}\right)^j N \leq \lambda^j(N) < \left(\frac{\theta_1}{\theta_2}\right)^j N + \theta_1\theta_2 \tag{5.2}$$

from which it is easy to see that the graphs $PSC(\lambda^j(N), \kappa, \theta_1, \theta_2)$ will indeed shrink exponentially in size, except for an additive adjustment, until we reach the base case $N \leq \theta_1\theta_2$.

Given the inequality (5.2), the calculation of the number of vertices and edges is just a straightforward recurrence. Also, note that once we know that the number of recursive steps is logarithmic in N, it immediately follows that the depth of the graph must be $\mathrm{O}(\log N)$ as well. We omit the details. □

Expressed in terms of the parameters in Definitions 5.7 and 5.8, [Pip77] established the existence of $(N, 6, 4, 6)$-linear concentrators and hence also of a family of superconcentrators $PSC(N, 6, 4, 6)$. These superconcentrators have $39N + \mathrm{O}(\log N)$ edges. The explicit construction in [GG81] yields superconcentrators $PSC(N, 112, 16, 17)$ with $3825N + \mathrm{O}(\sqrt{N})$ edges (under additional technical constraints such that $\lceil N/\theta_2 \rceil$ has to be a perfect square).

Recall from Theorem 5.6 that we want to find a black pebbling strategy for $PSC(N, \kappa, \theta_1, \theta_2)$ in simultaneous space $s = \Omega(\log N)$ and time $\mathrm{O}\big(N \cdot (N/s)^k\big)$ for some constant $k \in \mathbb{R}^+$. The exponent that we will get is $k = 1 + 2\log_{\theta_2/\theta_1} \kappa$.

Since the parameters θ_1, θ_2, and κ are all fixed in what follows, let us omit them in the notation and write just $PSC(N)$ for brevity. Also, let us write $RLC(N, j)$ to denote the subgraph of $PSC(N)$ obtained by recursively expanding Definition 5.8 j times, to get a sequence of j linear concentrators on the left and right, and then deleting the middle component $PSC(\lambda^j(N))$ as shown in Figure 13. Then $PSC(N)$ is the (edge-disjoint) union of $RLC(N, j)$ and $PSC(\lambda^j(N))$. Let us furthermore introduce the notation $\mathit{Left}(RLC(N, j))$ for the left half of the graph $RLC(N, j)$ and $\mathit{Right}(RLC(N, j))$ for the right half.

Algorithm 5.10 (PSC-pebble). Suppose that $s \geq K \log N$ for some large enough constant K. Define $j = j(N, s)$ to be the number of recursive steps needed to reach a subgraph $PSC(\lambda^j(N))$ with at most s vertices. Then the black pebbling strategy *PSC-pebble* for the Pippenger superconcentrators is as follows:

- If $j = j(N, s) = 0$, do the trivial pebbling of $PSC(N)$ in Observation 3.1 in time and space $\mathrm{O}(s)$.
- Otherwise, pebble $PSC(N)$ in three phases:
  1. Make a persistent pebbling of $\mathit{Left}(RLC(N, j))$ leaving pebbles on all sinks, i.e., on the sources of $PSC(\lambda^j(N))$.
  2. Do the trivial pebbling of $PSC(\lambda^j(N))$ leaving pebbles on all sinks, i.e., on the sources of $\mathit{Right}(RLC(N, j))$.
  3. Pebble all sinks of $PSC(N)$ (one by one, but not persistently) while keeping pebbles on all sources in $\mathit{Right}(RLC(N, j))$.

We now outline the argument that this pebbling strategy can be carried out within the time and space bounds stated in Theorem 5.6. The reader is referred to [LT82, Section 3] for details in the calculations omitted below.

By construction, phase 2 of Algorithm 5.10 can be done in time and space $\mathrm{O}(s)$. To analyze phases 1 and 3, we consider the unfolding of $RLC(N, j)$ into a family of trees as described in Definition 3.4 and Figure 3.

Look first at phase 1. Note that it follows from Lemma 5.9 and the inequality (5.2) that

$$j(N, s) \leq \log_{\theta_2/\theta_1}(N/s) + \mathrm{O}(1) \ . \tag{5.3}$$

Hence, $\mathit{Left}(RLC(N, j))$ has depth at most $\log_{\theta_2/\theta_1}(N/s) + \mathrm{O}(1)$. Also, by construction this graph has bounded indegree κ'. For a single-sink DAG G of depth d and indegree ℓ, it is easy to verify that $\mathit{unfold}(G)$ has less than $2\ell^d$ vertices. Therefore, for every sink in the subgraph $\mathit{Left}(RLC(N, j))$ we get a tree in $\mathit{unfold}\big(\mathit{Left}(RLC(N, j))\big)$ with $\mathrm{O}\big((\kappa')^j\big) = \mathrm{O}\left((\kappa\theta_2/\theta_1)^{\log_{\theta_2/\theta_1}(N/s)+\mathrm{O}(1)}\right) = \mathrm{O}\big((N/s)^{1+\log_{\theta_2/\theta_1}\kappa}\big)$ vertices. This yields the following bound for the first phase in *PSC-pebble*.

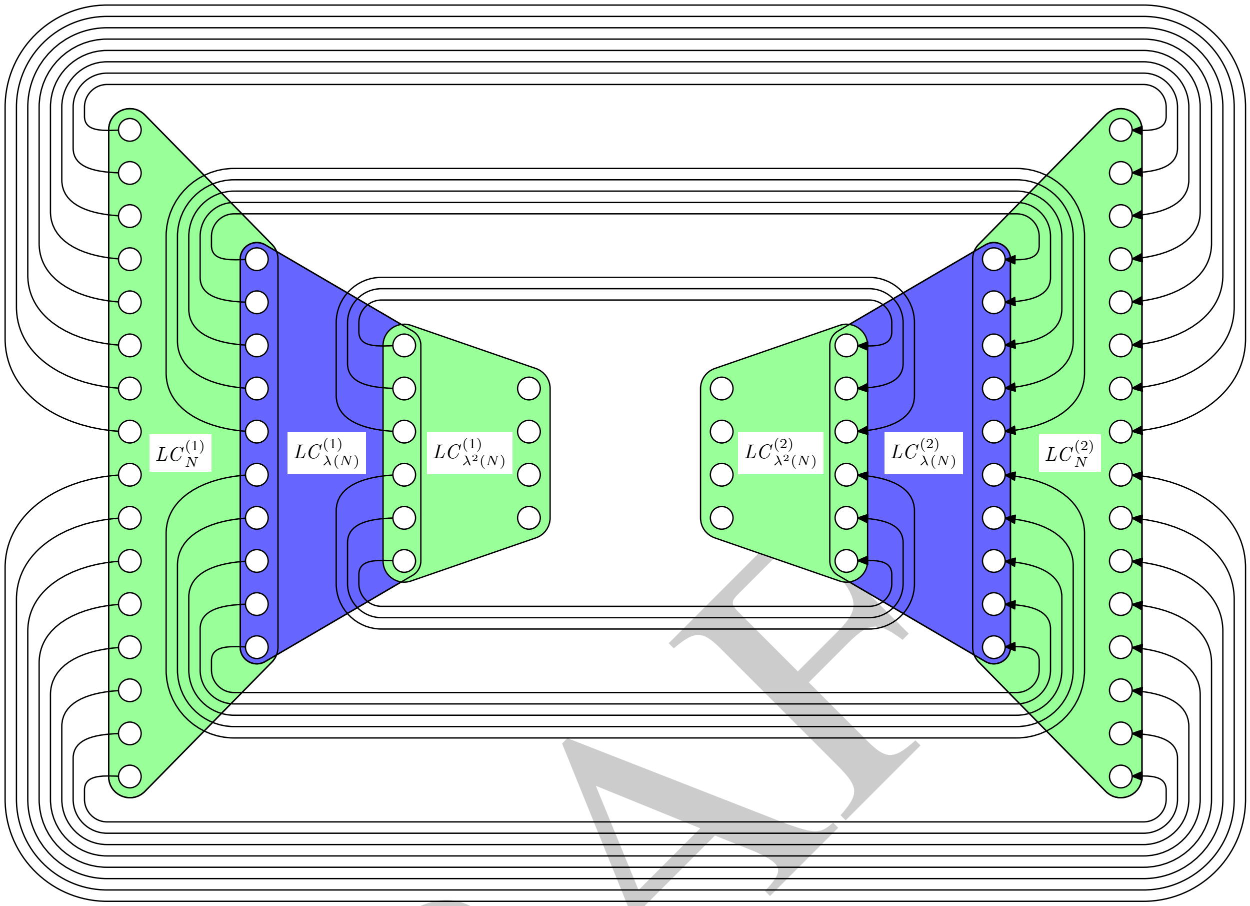


Figure 13: Graph $RLC(N, 3)$ consisting of three outer layers of $PSC(N, \kappa, \theta_1, \theta_2)$ recursively expanded.

Lemma 5.11. *Phase 1 in Algorithm 5.10 can be carried out in space s and time $\mathrm{O}\big(N \cdot (N/s)^{\log_{\theta_2/\theta_1} \kappa}\big)$.*

Proof. Consider the forest $unfold\big(Left(RLC(N, j))\big)$. By the choice of $j(N, s)$ we have that the graph $Left(RLC(N, j))$ has at most $s/2$ sinks, and hence $unfold\big(Left(RLC(N, j))\big)$ consists of at most $s/2$ trees. Apply the pebbling strategy in Observation 3.3 on each tree in this unfolding in some arbitrary order, leaving black pebbles on all sinks. Since the depth of $Left(RLC(N, j))$ is $\mathrm{O}(\log N)$ and $s \geq K \log N$ for K chosen large enough, in particular we can make sure that the pebbling space is at least twice the depth. This means that the strategy in Observation 3.3 pebbles each tree in $unfold\big(Left(RLC(N, j))\big)$ in linear time and with at most $s/2$ pebbles. Putting all of these pebblings together and appealing to Proposition 3.5, we get a persistent black pebbling of $Left(RLC(N, j))$ in space at most $s/2 + s/2 = s$ and time $\mathrm{O}\big(s \cdot (N/s)^{1+\log_{\theta_2/\theta_1} \kappa}\big) = \mathrm{O}\big(N \cdot (N/s)^{\log_{\theta_2/\theta_1} \kappa}\big)$. ☐

To bound the time and space of the third phase in the pebbling, we make a similar analysis but for all of $RLC(N, j)$. We start by estimating the number of vertices in $unfold\big(RLC(N, j)\big)$. In order not to get unnecessarily loose bounds we need to be slightly more careful here than in our calculations for $Left(RLC(N, j))$ in Lemma 5.11. To follow the proof of the next lemma, it is probably helpful to look back at Figures 12 and 13.

Lemma 5.12. *Consider $RLC(N, j)$ and let $L(j)$ be the maximum number of leaves and $V(j)$ be the maximum number of vertices in any tree in $unfold\big(RLC(N, j)\big)$. Then $L(0) = V(0) = 1$ and for $j > 0$ we* *have*

$$L(j) \le 1 + \kappa\kappa' \cdot L(j-1) \ ,$$
$$V(j) \le \kappa \cdot V(j-1) + \kappa\kappa' \cdot L(j-1) + 2 \ .$$

In particular, it holds that $L(j) = \mathrm{O}\big((\kappa\kappa')^j\big)$ *and* $V(j) = \mathrm{O}\big((\kappa\kappa')^j\big)$.

Proof sketch. For $j = 0$, $RLC(N, j)$ is just a collection of isolated vertices and so is $\mathit{unfold}\big(RLC(N, j)\big)$, so the claim in the lemma follows trivially.

Let $j > 0$ and consider the tree rooted in z_i in $\mathit{unfold}\big(RLC(N, j)\big)$. We get one leaf from the direct edge to z_i from the source s_i in $RLC(N, j)$. There are at most κ predecessors of z_i in $LC_N^{(2)}$. For each of these vertices we get at most $L(j-1)$ leaves in $\mathit{unfold}\big(RLC(\lambda(N), j-1)\big)$. For the leaves originating from sources in $\mathit{Right}(RLC(\lambda(N), j-1))$ on the right-hand side in Figure 13 we are done with the counting, but for the leaves on the left there are at most κ' predecessors in $LC_N^{(1)}$ that will become new leaves in the tree rooted in z_i in $\mathit{unfold}\big(RLC(N, j)\big)$. This gives the recurrence for $L(j)$.

For $V(j)$, we get two vertices from z_i and s_i. Again we have at most κ predecessors of z_i in $LC_N^{(2)}$. By induction we get at most $V(j-1)$ vertices from each predecessor, of which at most $L(j-1)$ are leaves in $\mathit{unfold}\big(RLC(\lambda(N), j-1)\big)$. From each leaf on the left in $\mathit{unfold}\big(RLC(\lambda(N), j-1)\big)$ we have a contribution of at most κ' vertices corresponding to sources in $LC_N^{(1)}$, and the recurrence for $V(j)$ follows.

Solving these recurrence relations as in [LT82, Theorem 3.3.10] yields the stated bounds. □

The analysis of phase 3 now becomes very similar to that of phase 1 in Lemma 5.11.

Lemma 5.13. *Any* r *sinks of the graph* $RLC(N, j)$ *can be pebbled in space* $s/2$ *and time* $\mathrm{O}\big(r \cdot (N/s)^k\big)$ *for* $k = 1 + 2\log_{\theta_2/\theta_1} \kappa$.

Proof. By Lemma 5.12, inequality (5.3), and our choice of $s = \Omega(\log N)$, the pebbling strategy in Observation 3.3 pebbles each tree in $\mathit{unfold}\big(RLC(N, j)\big)$ in space $s/2$ and time

$$\mathrm{O}\big((\kappa\kappa')^j\big) = \mathrm{O}\left((\kappa^2\theta_2/\theta_1)^{\log_{\theta_2/\theta_1}(N/s)+\mathrm{O}(1)}\right) = \mathrm{O}\big((N/s)^{1+2\log_{\theta_2/\theta_1}\kappa}\big) \ . \tag{5.4}$$

Multiplying by r and applying Proposition 3.5 gives the bound stated in the lemma. □

Combining Lemmas 5.11 and 5.13 we have the following theorem.

Theorem 5.14. *The black pebbling strategy PSC-pebble in Algorithm 5.10 pebbles any* r *sinks of* $PSC(N)$ *in space* s *and time* $\mathrm{O}\big(N \cdot (N/s)^{\log_{\theta_2/\theta_1}\kappa}\big) + \mathrm{O}\big(r \cdot (N/s)^{1+2\log_{\theta_2/\theta_1}\kappa}\big)$.

Proof. The first term corresponds to phase 1 and the second term to phase 3. The time required for phase 2 is dominated by that of phase 1. □

From this Theorem 5.6 follows as a corollary. More precisely, we can state the result as follows.

Corollary 5.15. *There is a black pebbling strategy that pebbles the linear superconcentrator* $PSC(N)$ *in space* s *and time* $\mathrm{O}\big(N \cdot (N/s)^k\big)$ *for* $k = 1 + 2\log_{\theta_2/\theta_1}\kappa$.

Proof. For $r = N$ in Theorem 5.14, phase 3 dominates the pebbling time. □

We remark that *PSC-pebble* can pebble the sinks in arbitrary order, so the same pebbling strategy also works for the single-sink version $\widehat{PSC(N)}$ (Definition 3.7) of the superconcentrator as well.

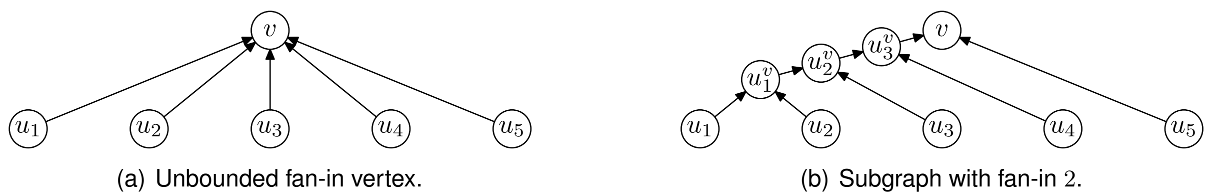


Figure 14: Conversion of unbounded indegree DAG to indegree 2 DAG.

5.4 Parameters for the Best Known Explicit Superconcentrator Construction

We conclude this section by stating the parameters for the explicit superconcentrator construction by Alon and Capalbo [AC03], which we will use (to have a fixed superconcentrator family for concreteness) later in this survey.

Theorem 5.16 ([AC03]). *For all $k \in \mathbb{N}$ there is an explicitly constructible superconcentrator $SC_{N(k)}$ with $N = N(k) = 4095 \cdot 2^k$ sources and sinks, $44N + \mathrm{O}(1)$ edges, $6N + \mathrm{O}(1)$ vertices, and fan-in 12.*

We remark that the construction in [AC03] works only for $k \geq 6$, but it is clear that we can get a statement on the form of Theorem 5.16 by complementing their superconcentrator family with some arbitrary hardcoded superconcentrators for $k < 6$ with a linear number of edges (or simply by using complete bipartite graphs with the appropriate modifications as described below to get the right bound on the fan-in).

For our applications of pebbling in proof complexity, we would like to have superconcentrators with minimal indegree 2. This is easy to take care of.

Corollary 5.17. *For all $k \in \mathbb{N}$ there is a superconcentrator $SC'_{N(k)}$ with $N = N(k) = 4095 \cdot 2^k$ sources and sinks, at most $66N + \mathrm{O}(1)$ vertices, and indegree 2.*

Proof. For every vertex v in the DAG SC_N in Theorem 5.16 with strictly more than 2 immediate predecessors $u_1, u_2, u_3, \ldots$, apply the local transformation from Figure 14(a) to Figure 14(b), where the u_i^v are new vertices unique for every v. It is easy to verify that the result after applying such a transformation is still a superconcentrator, since any pair of vertex-disjoint paths will remain vertex-disjoint in the new graph.

When we want to go from fan-in 12 to fan-in 2, a single fain-in 12 vertex leads to $(12 - 2) = 10$ new auxiliary vertices. Thus, all in all we get at most $10 \cdot (6N + \mathrm{O}(1))$ new vertices, plus the $6N + \mathrm{O}(1)$ vertices that we already had, for a total of $66N + \mathrm{O}(1)$. ☐

6 Two General Upper Bounds

For some applications (in proof complexity and elsewhere), one is interested in DAGs with as high a pebbling price as possible measured in terms of the number of vertices. What is the maximum number of pebbles needed to pebble a graph with n vertices? If there is no bound on the indegree, then clearly n pebbles can be required in the worst case, but if the indegree is bounded by some constant, Hopcroft et al. [HPV77] showed that the best one can hope for (or the worst one has to fear, depending on the perspective) is $\mathrm{O}(n/\log n)$.

Theorem 6.1 ([HPV77]). *For directed acyclic graphs G with n vertices and bounded indegree, the black pebbling price is at most $\mathsf{Peb}^{\emptyset}(G) = \mathrm{O}(n/\log n)$.*

As we shall see in Section 7, there are explicitly constructible graphs with asymptotically matching lower bounds on black-white pebbling price.

The space saving in Theorem 6.1 comes at an exponential increase in pebbling time, however. It is natural to ask whether this is necessary. Lengauer and Tarjan [LT82] proved the following upper bound on the time increase when optimizing space.

Theorem 6.2 ([LT82]). *For every directed acyclic graph G with n vertices and bounded indegree ℓ, and for every space parameter s satisfying $(3\ell + 2)n/\log n \leq s \leq n$, there is a (visiting) black pebbling strategy $\mathcal{P}$ for G with $\mathsf{space}(\mathcal{P}) \leq s$ and $\mathsf{time}(\mathcal{P}) \leq s \cdot 2^{2^{\mathrm{O}(n/s)}}$.*

This result, too, is asymptotically tight, and we will present correponding lower bounds in Section 11.

In this section we establish Theorems 6.1 and 6.2 using the very much simplified proofs by Loui [Lou80]. The only draw-back of Loui's proofs are that they contain a non-constructive step, whereas [HPV77, LT82] present explicit pebbling strategies. For the purposes of this survey, however, the existential statements in [Lou80] are enough.

6.1 Upper-bounding Pebbling Space in Terms of Internal Overlap

In the rest of this section, let G be a fixed graph with vertex set V, and let $W, W_1, W_2, \ldots$ denote subsets of vertices. We write $E(W_1, W_2) = \{(u, v) \in E(G) \mid u \in W_1, v \in W_2\}$ to denote the set of edges from W_1 to W_2 in G. The two key definitions in [Lou80] are as follows.

Definition 6.3 (Layered partition ([Lou80])). A *layered partition* of a vertex set W is a sequence of sets $(W_1, \ldots, W_m)$ such that W is the disjoint union of $W_1, \ldots, W_m$, and $E(W_j, W_i) = \emptyset$ for $j > i$.

Definition 6.4 (Internal overlap ([Lou80])). The *internal overlap* of a vertex set W is

$$\mathrm{io}(W) = \max\bigl\{|E(W_1, W_2)|\,; (W_1, W_2) \text{ is a layered partition of } W\bigr\} \ .$$

The internal overlap of the vertices of G provides an upper bound on the pebbling price of G.

Lemma 6.5 ([Lou80]). *There is a black pebbling strategy $\mathcal{P}$ for G in time $2n$ and space $\mathrm{io}(V) + 1$.*

Proof. The pebbling strategy is in n stages. Let W_i denote all vertices pebbled up to and including stage i, and define $W_0 = \emptyset$. At stage i, place a black pebble on some vertex $v \in V \setminus W_{i-1}$ such that there are black pebbles on $\mathit{pred}(v)$, and let $W_i = W_{i-1} \cup \{v\}$. Then remove pebbles from all vertices $u \in W_i$ such that $\mathit{succ}(u) \subseteq W_i$. By the black pebbling rules, $(W_i, V \setminus W_i)$ is a layered partition of V, and by construction $u \in W_i$ contains a pebble only if it is an immediate predecessor of some vertex in $V \setminus W_i$. But this proves that the number of pebbles at the end of stage i is at most $\bigl|E(W_i, V \setminus W_i)\bigr| \leq \mathrm{io}(V)$, and we only use one extra pebble at the start of each stage. Thus, the pebbling space is at most $\mathrm{io}(V)+1$. The time bound follows from observing that each vertex is pebbled and unpebbled exactly once. □

In fact, the pebbling in Lemma 6.5 can even be made persistent (provided that there are no isolated sinks in $Z(G)$ without incoming edges), since if (W_1, W_2) is a layered partition of the vertices in G, then so is $\bigl(W_1 \setminus Z(G), W_2 \cup Z(G)\bigr)$, and every pebble left on a sink corresponds to an edge crossing this partition. This observation does not help us much, however, since we cannot keep the pebbling persistent and linear-time when we generalize Lemma 6.5 to partitions containing two or more sets.

Lemma 6.6 ([Lou80]). *If $(W_1, \ldots, W_m)$ is a layered partition of the vertices of a DAG G with indegree ℓ, then there is a black pebbling strategy for G in space at most $\sum_{i=1}^{m}\bigl(\mathrm{io}(W_i) + \ell\bigr)$.*

Proof. By induction over m. The base case $m = 1$ follows from Lemma 6.5.

For the inductive step, let $\mathcal{P}_{m-1}$ be the complete black pebbling in space at most $\sum_{i=1}^{m-1}\bigl(\mathrm{io}(W_i)+\ell\bigr)$ of the subgraph induced by $W_1 \cup \ldots \cup W_{m-1} = V \setminus W_m$. Run the pebbling $\mathcal{P}'$ provided by Lemma 6.5 on the subgraph induced by W_m. Whenever $\mathcal{P}'$ tries to pebble a vertex $w \in W_m$ with immediate predecessors in $V \setminus W_i$, invoke the pebbling $\mathcal{P}_{m-1}$ once for every vertex $v \in \mathit{pred}(w) \cap (V \setminus W_m)$, leaving black pebbles on all such v. This requires space at most $(\ell - 1) + \sum_{i=1}^{m-1}\bigl(\mathrm{io}(W_i)+\ell\bigr)$ on the vertices in $V \setminus W_m$. As soon as a pebble has been place on $w \in W_m$, remove all pebbles from $V \setminus W_m$ again. Since $\mathcal{P}'$ never uses more than $\mathrm{io}(W_m) + 1$ pebbles on W_m, we get the space bound as stated in the lemma. □

The non-constructive part in Loui's proof is the assertion that there always exist layered partitions with good overlap properties.

Lemma 6.7 ([Lou80]). *For every DAG G with n vertices and indegree ℓ, and for every $r \in \mathbb{R}^+$, there is a layered partition $(W_1, \ldots, W_m)$ of V such that $m \leq 2^{\lceil \ell n/r \rceil}$ and $\sum_{i=1}^{m} \mathrm{io}(W_i) \leq \lfloor r \rfloor$.*

Proof. Suppose on the contrary that the lemma is false. Then for all layered partitions $(W_1, \ldots, W_m)$ of V with $m \leq 2^{\lceil \ell n/r \rceil}$ it holds that $\sum_{i=1}^{m} \mathrm{io}(W_i) > r$ (since the left-hand summation is an integer). Set $V_0^0 = V$ and find a layered partition $\bigl(V_0^1, V_1^1\bigr)$ of V_0^0 such that $\bigl|E(V_0^1, V_1^1)\bigr| = \mathrm{io}\bigl(V_0^0\bigr) > r$. Repeat this procedure inductively for $i = 1, 2, \ldots, \lceil \ell n/r \rceil - 1$ by splitting every V_j^i for $j = 0, 1, \ldots, 2^i - 1$ into a layered partition $\bigl(V_{2j}^{i+1}, V_{2j+1}^{i+1}\bigr)$ such that $\bigl|E\bigl(V_{2j}^{i+1}, V_{2j+1}^{i+1}\bigr)\bigr| = \mathrm{io}\bigl(V_j^i\bigr)$.

Then clearly for all i it holds that $\bigl(V_0^i, V_1^i, \ldots, V_{2^i-1}^i\bigr)$ is a layered partition of V and hence we have that $\sum_{j=0}^{2^i-1} \mathrm{io}\bigl(V_j^i\bigr) > r$ by assumption. Summing over all $i = 0, 1, \ldots, \lceil \ell n/r \rceil - 1$ we get

$$\sum_{i=0}^{\lceil \ell n/r \rceil - 1} \sum_{j=0}^{2^i-1} \mathrm{io}\bigl(V_j^i\bigr) > \lceil \ell n/r \rceil r \geq \ell n \ . \tag{6.1}$$

On the other hand, by construction all set of edges $E\bigl(V_{2j}^{i+1}, V_{2j+1}^{i+1}\bigr)$ are disjoint. To see this, note that for $k \leq i$ the kth partitions only count edges with at least one endpoint outside of $V_j^i = V_{2j}^{i+1} \cup V_{2j+1}^{i+1}$, and for $k > i + 1$ the subpartitions of V_{2j}^{i+1} and V_{2j+1}^{i+1} only count edges entirely within either V_{2j}^{i+1} or V_{2j+1}^{i+1}, respectively. Hence, the maximum possible number of edges ℓn in G must be an upper bound on the sum

$$\sum_{i=0}^{\lceil \ell n/r \rceil - 1} \sum_{j=0}^{2^i-1} \mathrm{io}\bigl(V_j^i\bigr) = \sum_{i=0}^{\lceil \ell n/r \rceil - 1} \sum_{j=0}^{2^i-1} \bigl|E\bigl(V_{2j}^{i+1}, V_{2j+1}^{i+1}\bigr)\bigr| \leq \ell n \ . \tag{6.2}$$

This is a contradiction, and the lemma follows. □

6.2 Proofs of Upper Bounds on Pebbling Price and Time-Space Trade-offs

Given the tools in Section 6.1, the proof of the $\mathrm{O}(n/\log n)$ upper bound on pebbling price in Theorem 6.1 is now just a matter of fixing a layered partition with the right overlap properties and then applying the pebbling strategy obtained from this layered partition.

Proof of Theorem 6.1. Let G be a DAG with n vertices and indegree ℓ. Set $s = 4\ell n/\log n$. We want to prove that $\mathsf{Peb}^{\emptyset}(G) \leq s$. We will make use of the inequality

$$\log\left(\frac{2n}{\log n}\right) = 1 + \log n - \log\log n \geq \left\lceil \frac{\log n}{2} \right\rceil \tag{6.3}$$

which can be verified to hold for all n (and, if nothing else, is easily seen to be true for n large enough).

According to Lemma 6.7 there is a layered partition $(W_1, \ldots, W_m)$ of V with $m \leq 2^{\lceil 2\ell n/s \rceil}$ and $\sum_{i=1}^{m} \mathrm{io}(W_i) \leq \lfloor s/2 \rfloor$. Then Lemma 6.6 says that there is a pebbling of G in space at most

$$\sum_{i=1}^{m} \bigl(\mathrm{io}(W_i) + \ell\bigr) = \left\lfloor \frac{s}{2} \right\rfloor + \ell m \leq s \tag{6.4}$$

where the inequality holds since

$$\begin{aligned} \ell m &\leq \ell \cdot 2^{\lceil 2\ell n/s \rceil} && \bigl[\text{ by Lemma 6.7 }\bigr] \\ &\leq \ell \cdot 2^{\lceil (\log n)/2 \rceil} && \bigl[\text{ since } s = 4\ell n/\log n\bigr] \\ &\leq \ell \frac{2n}{\log n} = s/2 && \bigl[\text{ by the inequality (6.3) }\bigr] \end{aligned}$$

and the theorem follows. □

The proof of the upper bound on the time-space trade-off in Theorem 6.2 requires a bit more work, but the general idea is the same.

Proof of Theorem 6.2. Suppose that $(3\ell + 2)n/\log n \leq s \leq n$. Set

$$\alpha = \frac{2}{3} \tag{6.5}$$

$$\beta = \frac{\ell}{3\ell + 2} \tag{6.6}$$

$$\gamma = 1 - \alpha - \beta \tag{6.7}$$

and note that $\gamma > 0$. Let us first establish a couple of useful inequalities. Evidently,

$$\log\log n \leq \frac{1}{2} \log n \tag{6.8}$$

because $(3\ell + 2)n/\log n \leq n$ implies $n \geq 16$. Using this we conclude that

$$\begin{aligned} \log\bigl(\beta s/\ell\bigr) &\geq \log\bigl(n/\log n\bigr) \\ &\geq \frac{1}{2} \log n \\ &= \frac{3\ell n}{2} \cdot \frac{\log n}{(3\ell + 2)n} + n \cdot \frac{\log n}{(3\ell + 2)n} \\ &\geq \frac{\ell n}{\alpha s} + \frac{n}{s} \\ &\geq \frac{\ell n}{\alpha s} + 1 \end{aligned} \tag{6.9}$$

from which it follows that

$$\ell \cdot 2^{\lceil \ell n/\alpha s \rceil} \leq \beta s \ . \tag{6.10}$$

Apply Lemma 6.7 with $r = \alpha s$. This yields a layered partition $(W_1, \ldots, W_m)$ with at most

$$m \leq 2^{\lceil \ell n/\alpha s \rceil} \leq s/(3\ell + 2) \tag{6.11}$$

vertex sets and total overlap at most

$$\sum_{i=1}^{m} \mathrm{io}(W_i) \leq \lfloor \alpha s \rfloor \ . \tag{6.12}$$

Now for each $i = 1, \ldots, m$ we create three piles of pebbles of sizes

$$P_i = \mathrm{io}(W_i) \tag{6.13}$$
$$Q_i = \ell \tag{6.14}$$
$$R_i = \lfloor \gamma s |W_i| / n \rfloor \tag{6.15}$$

which is in order since the total number of pebbles will be

$$\begin{aligned} \sum_{i=1}^{m} (P_i + Q_i + R_i) &\le \sum_{i=1}^{m} \mathrm{io}(W_i) \; + \; \ell m \; + \; \gamma s \sum_{i=1}^{m} |W_i| / n \\ &\le \alpha s + \frac{\ell s}{3\ell + 2} + \gamma s \\ &\le (\alpha + \beta + \gamma) s = s \; . \end{aligned} \tag{6.16}$$

Note that $\sum_{i=i}^{m} (P_i + Q_i)$ pebbles are enough to carry out the pebbling strategy $\mathcal{P}_m$ in Lemma 6.6. We want to use the extra pebbles in R_i to speed up this pebbling a bit by avoiding excessive repebbling in the recursive invocations of subpebblings in Lemma 6.6.

We define the modified pebbling strategies $\mathcal{P}_j^*$ inductively and let $T(j)$ denote the time required for carrying out the pebbling strategy $\mathcal{P}_j^*$ on the subgraph induced by $W_1 \cup \ldots \cup W_j$. For $j = 1$, the pebbling in Lemma 6.5 uses at most $\mathrm{io}(W_1) + 1 \le P_1 + Q_1$ pebbles and runs in time $2 \cdot |W_1| \le 2n$.

Suppose we have a pebbling $\mathcal{P}_{m-1}^*$ in time $T(m-1)$ for the subgraph induced by $W_1 \cup \ldots \cup W_{m-1}$. To pebble W_m, do as in Lemma 6.6 but with the following modification. Run the pebbling strategy $\mathcal{P}'$ from Lemma 6.5 on W_m. When a pebble is needed on some predecessor in $W_1 \cup \ldots \cup W_{m-1}$, invoke $\mathcal{P}_{m-1}^*$ but use the $Q_m + R_m$ pebbles to leave black pebbles simultaneously on all immediate predecessors in $W_1 \cup \ldots \cup W_{m-1}$ of the $\lfloor (Q_m + R_m)/\ell \rfloor$ vertices in W_m that are next in turn to be pebbled by $\mathcal{P}'$. The number of times $\mathcal{P}'$ invokes $\mathcal{P}_{m-1}^*$ will then be at most

$$\left\lceil \frac{|W_m|}{\lfloor (Q_m + R_m)/\ell \rfloor} \right\rceil = \left\lceil \frac{|W_m|}{\lfloor (\ell + \lfloor \gamma s |W_m| / n \rfloor)/\ell \rfloor} \right\rceil \le \left\lceil \frac{\ell n}{\gamma s} \right\rceil \le 1 + \frac{\ell n}{\gamma s} \tag{6.17}$$

and it follows that an upper bound on the time required for $\mathcal{P}_m^*$ is

$$\begin{aligned} T(m) &\le \left\lceil \frac{|W_m|}{\lfloor (Q_m + R_m)/\ell \rfloor} \right\rceil T(m-1) + 2 \cdot |W_m| \\ &\le \left(1 + \frac{\ell n}{\gamma s}\right) T(m-1) + 2n \\ &\le 2n \sum_{i=0}^{m-1} \left(1 + \frac{\ell n}{\gamma s}\right)^i \\ &\le 2n \frac{(1 + \ell n/\gamma s)^m}{\ell n / \gamma s} \\ &= \frac{2\gamma}{\ell} s \cdot 2^{m \log(1 + \ell n/\gamma s)} \\ &\le \frac{2\gamma}{\ell} s \cdot 2^{2^{\lceil \ell n/\alpha s \rceil + \log\log(1+\ell n/\gamma s)}} \\ &\le s \cdot 2^{2^{\mathrm{O}(n/s)}} \end{aligned} \tag{6.18}$$

which proves the theorem. $\square$

7 An Optimal Lower Bound on Pebbling Price

In this section we present the result by Gilbert and Tarjan [GT78] that the $O(n/\log n)$ upper bound on pebbling price in Theorem 6.1 is asymptotically tight for black-white pebbling, and thus for black pebbling as well.

Theorem 7.1 ([GT78]). *There is a family of explicitly constructible DAGs $\{G_n\}_{n=1}^{\infty}$ with $\Theta(n)$ vertices, unique sink, and indegree 2 such that* $\textsf{BW-Peb}(G) = \Omega(n/\log n)$.

Gilbert and Tarjan use a family of graphs devised by Paul et al. [PTC77], who proved the corresponding lower bound for pebbling with black pebbles only. We remark that no explicit graph constructions were known at the time of the original theorems in [GT78, PTC77]. What is needed are explicitly constructible superconcentrators of constant density, and as was mentioned in Section 5 it has since been shown how to build such graphs.

7.1 Definition of Graph Family and Main Technical Lemma

Let us start by describing the graph family used in [GT78, PTC77], which we give the provisional name of *Gilbert-Tarjan graphs.* A more appetizing way of digesting Definition 7.2 below is perhaps simply to have a look at Figure 15 .

Definition 7.2 (Gilbert-Tarjan graphs ([GT78, PTC77])). Let $C(k) = SC_{N(k)}$ for $k = 0, 1, 2, \ldots$ denote any arbitrary but fixed family of superconcentrators with $N(k) = K \cdot 2^k$ sources and sinks for some constant $K \in \mathbb{N}^+$ and $\Theta(N(k))$ vertices of indegree 2. Then the Gilbert-Tarjan graph $\Xi(0)$ is $C(0)$, and $\Xi(i+1)$ for $i \geq 0$ is defined inductively as follows.

The graph $\Xi(i+1)$ has sources $s_{i+1}[j]$ and sinks $z_{i+1}[j]$ for $j = 1, 2, \ldots, N(i+1)$. It contains two copies $\Xi^{(1)}(i), \Xi^{(2)}(i)$ of the Gilbert-Tarjan graph of one size smaller with sources $s_i^{(c)}[j]$ and sinks $z_i^{(c)}[j]$ for $j = 1, 2, \ldots, N(i)$ and $c = 1, 2$, and two superconcentrator copies $C^{(1)}(i), C^{(2)}(i)$ with sources $x_i^{(c)}[j]$ and sinks $y_i^{(c)}[j]$ for $j = 1, 2, \ldots, N(i)$ and $c = 1, 2$. The edges in $\Xi(i+1)$ are all internal edges within $\Xi^{(1)}(i), \Xi^{(2)}(i)$ and $C^{(1)}(i), C^{(2)}(i)$, as well as the following edges:

1. $\bigl(s_{i+1}[j], x_i^{(1)}[j]\bigr)$ and $\bigl(s_{i+1}[j+N(i)], x_i^{(1)}[j]\bigr)$ for $j = 1, \ldots, N(i)$, from the sources in $\Xi(i+1)$ to the sources of $C^{(1)}(i)$,
2. $\bigl(y_i^{(1)}[j], s_i^{(1)}[j]\bigr)$ for $j = 1, \ldots, N(i)$, from the sinks of $C^{(1)}(i)$ to the sources of $\Xi^{(1)}(i)$,
3. $\bigl(z_i^{(1)}[j], s_i^{(2)}[j]\bigr)$ for $j = 1, \ldots, N(i)$, from the sinks of $\Xi^{(1)}(i)$ to the sources of $\Xi^{(2)}(i)$,
4. $\bigl(z_i^{(2)}[j], x_i^{(2)}[j]\bigr)$ for $j = 1, \ldots, N(i)$, from the sinks of $\Xi^{(2)}(i)$ to the sources of $C^{(2)}(i)$,
5. $\bigl(y_i^{(2)}[j], z_{i+1}[j]\bigr)$ and $\bigl(y_i^{(2)}[j], z_{i+1}[j+N(i)]\bigr)$ for $j = 1, \ldots, N(i)$, from the sinks of $C^{(2)}(i)$ to the sinks of $\Xi(i+1)$,
6. $\bigl(s_{i+1}[j], z_{i+1}[j]\bigr)$ for $j = 1, \ldots, N(i+1)$, directly from the sources to the sinks of $\Xi(i+1)$.

Since the size of the Gilbert-Tarjan graphs satisfy the recurrence $S(i+1) = 2S(i) + \Theta(2^i)$, we have the next proposition.

Proposition 7.3 ([GT78, PTC77]). *The graphs $\Xi(i)$ have $\Theta\bigl(2^i\bigr)$ sources and sinks, and are of size $\Theta\bigl(i \cdot 2^i\bigr)$ and indegree 2.*

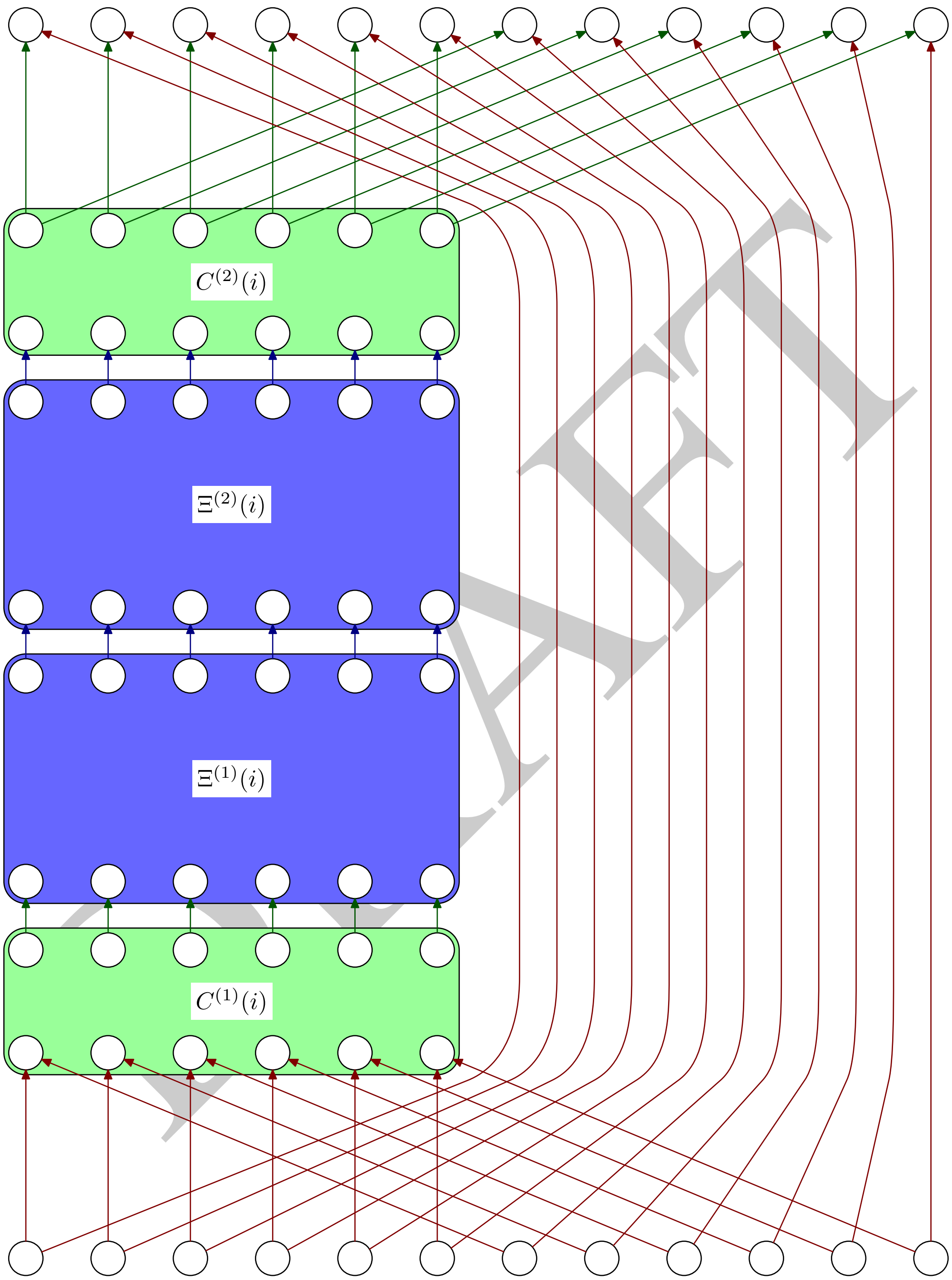


Figure 15: Construction in [GT78, PTC77] of DAG with maximal pebbling price in terms of size.

When stating the main technical lemma in [GT78], we assume for simplicity that $K \geq 1260$ in Definition 7.2. This is true for the superconcentrators in Corollary 5.17 that we will soon plug in there, and for other superconcentrators we can just skip the first members of $\Xi(i)$ until the number of sources and sinks reaches this number. The notation

$$M(i) = N(i)/N(0) \tag{7.1}$$

will be used extensively below as a convenient shorthand.

Lemma 7.4 (Main lemma ([GT78])). *Let $\Xi(i)$ be a DAG as in Definition 7.2 with $K \geq 1260$. Suppose $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling on $\Xi(i)$ such that*

- $\max\{\mathit{space}(\mathbb{P}_\sigma), \mathit{space}(\mathbb{P}_\tau)\} \leq 3 \cdot M(i)$, *and*
- *$\mathcal{P}$ pebbles at least $80 \cdot M(i)$ sinks in $\Xi(i)$ during $[\sigma, \tau]$.*

Then there is a subinterval $[\sigma', \tau'] \subseteq [\sigma, \tau]$ such that

- $\min_{t \in [\sigma', \tau']}\{\mathit{space}(\mathbb{P}_t)\} \geq M(i)$, *i.e., there are at least $M(i)$ pebbles on the DAG throughout the whole interval $[\sigma', \tau']$, and*
- *$\mathcal{P}$ pebbles at least $180 \cdot M(i)$ sources in $\Xi(i)$ during $[\sigma', \tau']$.*[2]

Postponing the proof of the lemma for a moment, however, let us first see how it yields Theorem 7.1.

Proof of Theorem 7.1. Lemma 7.4 clearly implies that $\mathit{BW\text{-}Peb}^{\emptyset}(\Xi(i)) = \Omega(2^i)$ since a complete pebbling has $\mathbb{P}_\sigma = \mathbb{P}_\tau = (\emptyset, \emptyset)$ and pebbles all sinks, not just a $80N(i)/N(0)$ fraction of them.

Let $\Xi(i)$ be the Gilbert-Tarjan graphs built on the explicit superconcentrators in Corollary 5.17, and define the graph family $\{H_n\}_{n=1}^{\infty}$ by $H_n = \Xi(\lfloor \log n - \log\log n \rfloor)$. Then $\mathit{BW\text{-}Peb}^{\emptyset}(H_n) = \Omega(n/\log n)$ by Lemma 7.4, and H_n has size $\Theta(n)$ by Proposition 7.3.

Finally, let $G_n = \widehat{H_n}$ be the single-sink version of H_n (Definition 3.7). Then by Observation 3.8 we have $\mathit{BW\text{-}Peb}(G_n) = \Omega(n/\log n)$, and $\{G_n\}_{n=1}^{\infty}$ is a family of explicitly constructible DAGs with $\Theta(n)$ vertices, unique sink, and indegree 2. □

7.2 Proof of the Optimal Lower Bound on Pebbling Price

In this subsection, we prove Lemma 7.4. Suppose we have Gilbert-Tarjan graphs $\Xi(i)$ with the base case graph $\Xi(0)$ having at least 1260 sources and sinks (i.e., $K \geq 1260$ in Definition 7.2). We will refer a number of times to Figure 16, which shows the Gilbert-Tarjan graphs with one level of the recursion unfolded. Hopefully, it will be easier to see the truth of some of the claims made below by studying this picture.

It is easy to see that any pebbling $\mathcal{P}$ as in Lemma 7.4 must pebble many sources. In fact, it is not hard to see from Figure 16 that $\Xi(i)$ is a superconcentrator itself, so from the Basic Lower Bound Argument in Lemma 5.4 it follows that $\mathcal{P}$ must pebble almost all sources in $\Xi(i)$. The difficult part is to prove that a substantial fraction of the sources are pebbled during an interval when there are many pebbles on the DAG.

Let us try to describe the roadmap for the proof. Since the graph construction is inductive, the proof is also by induction. Suppose we know that the lemma holds for $\Xi(i)$. The statement for $\Xi(i+1)$ just multiplies everything by a factor of 2, so if we could somehow apply the induction hypothesis simultaneously to both subgraphs $\Xi^{(1)}(i)$ and $\Xi^{(2)}(i)$, we should be in good shape. But to use induction on $\Xi^{(1)}(i)$ and $\Xi^{(2)}(i)$,

[2] We note that we have made no attempt to optimize the multiplicative constants in the lemma, but instead have chosen the figures to make the calculations as clean as possible. The original paper [GT78] has 49 instead of 80, 110 instead of 180, and $K \geq 1024$ instead of $K \geq 1260$ in Lemma 7.4, and the proof we are going to present below can be verified to work for these constants as well.

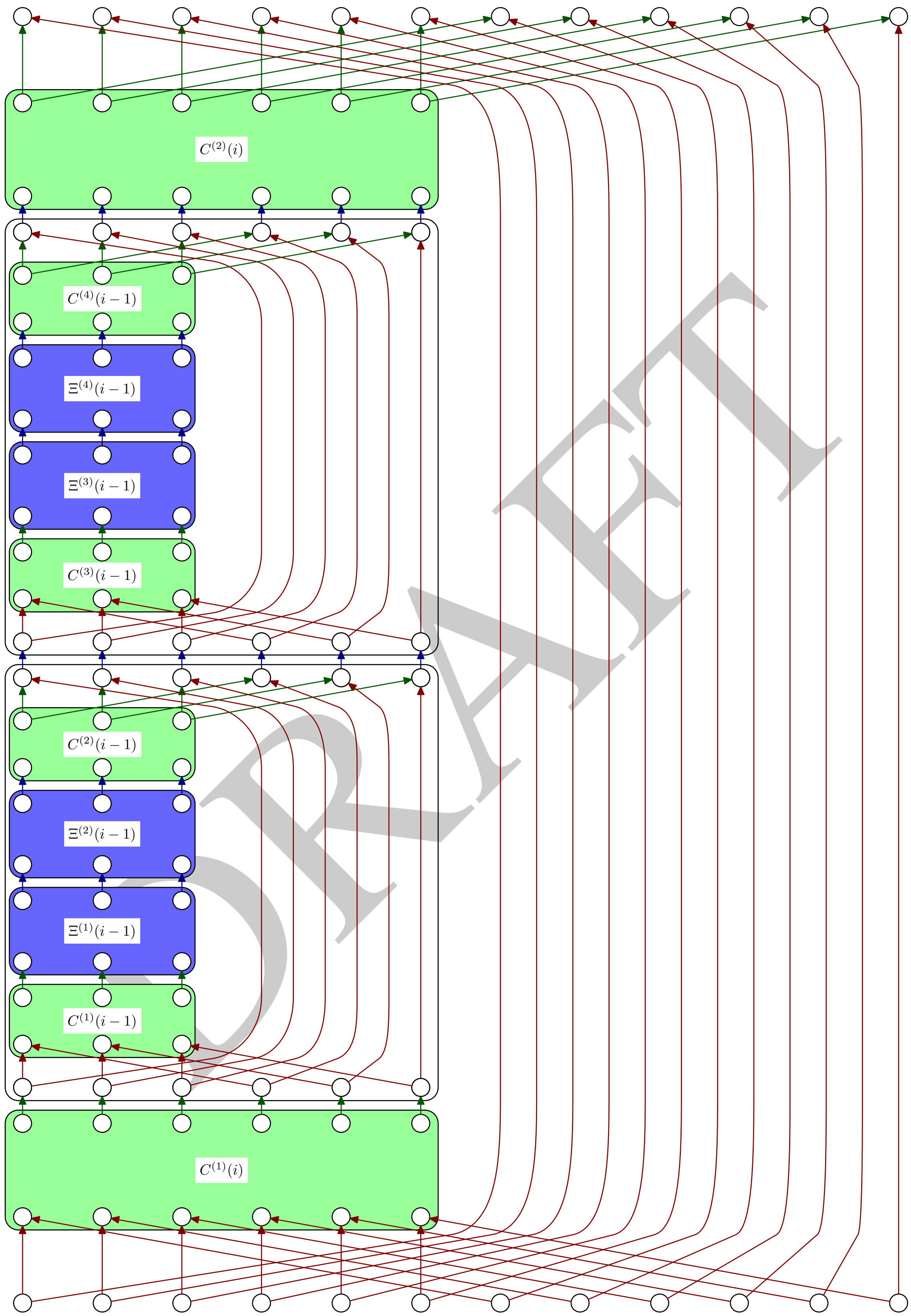


Figure 16: DAG $\Xi(i+1)$ in [GT78, PTC77] drawn in terms of $\Xi(i-1)$, $C(i-1)$ and $C(i)$.

we need an upper bound $3 \cdot M(i)$ on the number of pebbles at the start and the end of the subpebblings, and the statement of the lemma only guarantees an uper bound of $3 \cdot M(i+1) = 6 \cdot M(i)$ pebbles.

Suppose therefore as a special case that there are at least $3 \cdot M(i)$ pebbles on $\Xi(i+1)$ during an interval when a substantial fraction of the sinks of $\Xi(i+1)$ are pebbled, meaning that an appeal to the induction hypothesis is ruled out. But then $3 \cdot M(i) > M(i+1)$ is certainly a large enough number of pebbles to meet the requirements in the conclusion of the lemma anyway, and it turns out that we can use the superconcentrators in $\Xi(i+1)$ and the BLBA-lemma 5.4 to argue that sufficiently many sources in $\Xi(i+1)$ are pebbled during this interval for the inductive step to go through. In the same way, we can argue that if there are at least $3 \cdot M(i)$ pebbles on $\Xi(i+1)$ during an interval when substantial pebbling progress is made on $\Xi^{(1)}(i)$ or $\Xi^{(2)}(i)$, then Lemma 7.4 follows simply from the expansion properties of the superconcentrators in $\Xi(i+1)$, without any need for the induction hypothesis. Hence, for these special cases we are already home.

Having considered these special cases, we can then argue (with a bit of care) that if none of the special cases hold, there must exist an interval such that the prerequisites of Lemma 7.4 are met for both $\Xi^{(1)}(i)$ and $\Xi^{(2)}(i)$ simultaneously. This gives us $M(i)$ pebbles each on these two subgraphs, summing up to $M(i+1)$, and by a concluding argument we deduce that the fact that many sources are pebbled in $\Xi^{(1)}(i)$ must imply that many sources are pebbled in $\Xi(i+1)$ as well.

We now formalize this proof sketch. The following proposition takes care of the base case of the induction.

Proposition 7.5 (Base case). *Let $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ be a pebbling of $\Xi(0)$ such that there are at most 3 pebbles on $\Xi(0)$ at times σ and τ and $\mathcal{P}$ pebbles 80 sinks during $[\sigma, \tau]$. Then there is a subinterval $[\sigma', \tau'] \subseteq [\sigma, \tau]$ such that there is at least one pebble on $\Xi(0)$ during all of $[\sigma', \tau']$ and $\mathcal{P}$ pebbles at least 180 sources in $\Xi(0)$ during this interval.*

Proof. There are at most 3 vertices having pebbles at time σ and at most 3 other vertices having pebbles at time τ, so by the Lemma 5.3 there is some set of $7 \leq 80$ of the pebbled sinks that are reachable by paths from at least $N(0) - 6$ sources such that these paths are pebble-free at times σ and τ. At least one of the sinks z^* has $\lceil (N(0) - 6)/7 \rceil \geq 180$ such sources. Suppose z^* is pebbled at time t^*. Let $\sigma' - 1 \leq t^*$ be the last time when all 180 paths to z^* are pebble-free and $\tau' + 1 \geq t^*$ be the first time when all 180 paths to z^* are pebble-free again. Then by Lemma 3.11 it must hold that during the interval $[\sigma', \tau']$ at least 180 sources in $\Xi(0)$ are pebbled while there is at least one pebble on the graph. ☐

Next, we study the special cases mentioned in the proof outline above.

Lemma 7.6 (Special case 1). *Let $\mathcal{P}$ be a pebbling of $\Xi(i+1)$ meeting the prerequisites of Lemma 7.4, and suppose there is a time interval $[\sigma^*, \tau^*] \subseteq [\sigma, \tau]$ such that at least $45 \cdot M(i)$ sources of the subgraph $\Xi^{(1)}(i)$ are pebbled while there are at least $3 \cdot M(i)$ pebbles on $\Xi(i+1)$ during the whole interval.*

Then the conclusions in Lemma 7.4 hold. That is, there is a subinterval $[\sigma', \tau'] \subseteq [\sigma, \tau]$ such that $\min_{t \in [\sigma', \tau']}\{\mathsf{space}(\mathbb{P}_t)\} \geq M(i+1)$ and $\mathcal{P}$ pebbles at least $180 \cdot M(i+1)$ sources in $\Xi(i+1)$ during $[\sigma', \tau']$.

Proof. Let H'_L be the subgraph induced on $C^{(i)}(1)$ plus the sources of $\Xi^{(1)}(i)$ and the left-hand half of the sources of $\Xi(i+1)$, i.e., the vertices $\{s_{i+1}[j],\, s_i^{(1)}[j] \mid j \in [N(i)]\}$. Then H'_L is just $C^{(i)}(1)$ plus the added edges $\{(s_{i+1}[j], x_i^{(1)}[j]) \mid j \in [N(i)]\}$ to the sources of $C^{(i)}(1)$ and $\{(y_i^{(1)}[j], s_i^{(1)}[j]) \mid j \in [N(i)]\}$ from the sinks of $C^{(i)}(1)$, so H'_L is clearly a superconcentrator. Define H'_R in the same way except that it uses the right-hand half $\{s_{i+1}[j + N(i)] \mid j \in [N(i)]\}$ of the sources of $\Xi(i+1)$ instead of the left-hand half, and note that H'_R is a superconcentrator as well.

Let $\sigma^* < \sigma'$ be the last time when there are at most $3 \cdot M(i+1)$ pebbles on $\Xi(i+1)$ and let $\tau^* > \tau'$ be the first time when there are at most $3 \cdot M(i+1)$ pebbles on $\Xi(i+1)$ again. Note that σ^* and τ^* must exist since the endpoints σ and τ of the whole interval satify these conditions by the assumptions in Lemma 7.4. Furthermore, since every move changes the pebbling space by it most one, it holds that there are at least $3 \cdot M(i) - 1 \geq M(i+1)$ pebbles on $\Xi(i+1)$ during the whole interval $[\sigma^*, \tau^*]$. We claim that at least $180 \cdot M(i+1)$ sources in $\Xi(i+1)$ must be pebbled during $[\sigma^*, \tau^*]$, which yields the lemma.

To establish the claim, note that it is sufficient to prove that at least $180 \cdot M(i)$ sources are pebbled in each of the graphs H'_L and H'_R defined above. Consider H'_L. The sinks of H'_L are the sources of $\Xi^{(1)}(i)$, so at least $45 \cdot M(i)$ sinks in H'_L gets pebbled during $[\sigma^*, \tau^*]$ Since $45 \cdot M(i) > 2 \cdot 3 \cdot M(i+1) \geq \mathit{space}(\mathbb{P}_{\sigma^*}) + \mathit{space}(\mathbb{P}_{\tau^*})$, the BLBA-lemma 5.4 says that at least $N(i) - 6 \cdot M(i+1)$ sources in H'_L are pebbled during $[\sigma^*, \tau^*]$. Summing with the sources from H'_R obtained by completely analogous reasoning, we see that the total number of pebbled sources is at least

$$2 \cdot N(i) - 12 \cdot M(i+1) = \big(N(0) - 12\big) M(i+1) \geq 180 \cdot M(i+1) \tag{7.2}$$

and the lemma follows. □

We have exactly the same statement as in Lemma 7.6 with respect to the sources of the second copy $\Xi^{(2)}(i)$ of the Gilbert-Tarjan subgraph as well.

Lemma 7.7 (Special case 2). *Let $\mathcal{P}$ be a pebbling of $\Xi(i+1)$ meeting the prerequisites of Lemma 7.4, and suppose there is an interval $[\sigma^*, \tau^*] \subseteq [\sigma, \tau]$ such that at least $45M(i)$ sources of $\Xi^{(2)}(i)$ are pebbled while there are at least $3 \cdot M(i)$ pebbles on $\Xi(i+1)$. Then the conclusions in Lemma 7.4 hold.*

Proof. Again define a subgraph H''_L which is H'_L as constructed in the proof of Lemma 7.6 plus the sinks of $\Xi^{(1)}(i)$ and the sources of $\Xi^{(2)}(i)$, as well as the direct edges $\big\{\big(s_i^{(1)}[j], z_i^{(1)}[j]\big) \bigm| j \in [N(i)]\big\}$ bypassing $\Xi^{(1)}(i)$ and the edges $\big\{\big(z_i^{(1)}[j], s_i^{(2)}[j]\big) \bigm| j \in [N(i)]\big\}$ connecting $\Xi^{(1)}(i)$ and $\Xi^{(2)}(i)$. Let H''_R be the analogous graph induced on the the right-hand half $\big\{s_{i+1}[j + N(i)] \bigm| j \in [N(i)]\big\}$, of the sources of $\Xi(i+1)$ instead of the left-hand half.

Studying Figure 16, it is easy to see that H''_L and H''_R are both just $C^{(1)}(i)$ with some disjoint paths added to the sources and sinks, and so are clearly superconcentrators with the sources of $\Xi^{(2)}(i)$ as sinks. Now we can reuse the second part of the proof of Lemma 7.6 word by word. □

The third special case is an analogous statement but with respect not to the sources of the subgraphs $\Xi^{(c)}(i)$ but with respect to the sinks of $\Xi(i+1)$.

Lemma 7.8 (Special case 3). *Let $\mathcal{P}$ be a pebbling of $\Xi(i+1)$ meeting the prerequisites of Lemma 7.4, and suppose there is an interval $[\sigma^*, \tau^*] \subseteq [\sigma, \tau]$ such that at least $20 \cdot M(i+1)$ sinks of $\Xi(i+1)$ are pebbled while there are at least $3 \cdot M(i)$ pebbles on $\Xi(i+1)$. Then the conclusions in Lemma 7.4 hold.*

Proof. There are at least $10 \cdot M(i)$ sinks pebbled either in the left-hand half $\big\{z_{i+1}[j] \bigm| j \in [N(i)]\big\}$ of the sinks or in the right-hand half $\big\{z_{i+1}[j+N(i)] \bigm| j \in [N(i)]\big\}$. Suppose without loss of generality that $10 \cdot M(i)$ sinks are pebbled in the left-hand half.

Let H'''_L be the subgraph H''_L constructed in the proof of Lemma 7.7 plus the sinks of $\Xi^{(2)}(i)$, the superconcentrator $C^{(2)}(i)$, and the left-hand side $\big\{z_{i+1}[j] \bigm| j \in [N(i)]\big\}$ of the sinks of $\Xi(i+1)$. Add to the edges of H''_L and $C^{(2)}(i)$ also the direct edges $\big\{\big(s_i^{(2)}[j], z_i^{(2)}[j]\big) \bigm| j \in [N(i)]\big\}$ bypassing $\Xi^{(1)}(i)$, the edges $\big\{\big(z_i^{(2)}[j], x_i^{(2)}[j]\big) \bigm| j \in [N(i)]\big\}$ connecting $\Xi^{(2)}(i)$ and $C^{(2)}(i)$, and the edges $\big\{\big(y_i^{(2)}[j], z_{i+1}[j]\big) \bigm| j \in [N(i)]\big\}$ connecting $C^{(2)}(i)$ with the left-hand side of the sinks of $\Xi(i+1)$. Again, it should hopefully be easy by looking at Figure 16 to figure out what H'''_L looks like. We let H'''_R be the analogous graph using the right-hand half of the sources of $\Xi(i+1)$ instead of the left-hand half.

As in the proof of Lemma 7.6, find an interval $[\sigma^*, \tau^*] \supseteq [\sigma, \tau]$ such that there are at least $3 \cdot M(i) - 1 \geq M(i+1)$ pebbles on $\Xi(i+1)$ during the whole interval $[\sigma^*, \tau^*]$ but at most $3 \cdot M(i+1)$ pebbles at times σ^* and τ^*.

Since $10 \cdot M(i+1) > 6 \cdot M(i+1)$, the BLBA-lemma 5.4 applied twice says that a total of at least $2 \cdot N(i) - 12 \cdot M(i+1) = \bigl(N(0) - 12\bigr) M(i+1) \geq 180 \cdot M(i+1)$ sources in H'''_L and H'''_R are pebbled during $[\sigma^*, \tau^*]$, and the lemma follows. □

We remark that so far, we have not used the second superconcentrator $C^{(2)}(i)$ at all. In the proof of Lemma 7.8 we could have defined H'''_L and H'''_R by just fixing some vertex-disjoint paths through $C^{(2)}(i)$. All claims made so far have followed from the expansion properties of $C^{(1)}(i)$. Using the special case lemmas Lemmas 7.6, 7.7, and 7.8, we now take care of the general case where both the induction hypothesis and the superconcentrator $C^{(2)}(i)$ are needed.

Lemma 7.9 (General case). *Let $\mathcal{P}$ be a pebbling of $\Xi(i+1)$ meeting the prerequisites of Lemma 7.4, and suppose none of the cases in Lemmas 7.6, 7.7, or 7.8 apply. Then the conclusions in Lemma 7.4 also hold, i.e., there is a subinterval $[\sigma', \tau'] \subseteq [\sigma, \tau]$ such that $\min_{t \in [\sigma', \tau']} \{\mathit{space}(\mathbb{P}_t)\} \geq M(i+1)$ and $\mathcal{P}$ pebbles at least $180 \cdot M(i+1)$ sources in $\Xi(i+1)$ during $[\sigma', \tau']$.*

Proof. The idea in the proof is to create nested subintervals $[\sigma, \tau] \supseteq [\sigma_1, \tau_1] \supseteq [\sigma_2, \tau_2] \supseteq [\sigma_3, \tau_3] \supseteq \ldots$ such that we will be able to apply the induction hypothesis to both subgraphs $\Xi^{(1)}(i)$ and $\Xi^{(2)}(i)$ simultaneously inside these nested interval. We define the intervals step by step.

1. Consider the time interval when the first quarter of the sinks in $\Xi(i+1)$ are pebbled. At some point in time during this interval the space must go below $3 \cdot M(i)$ pebbles, for otherwise we would be in the special case 3 of Lemma 7.8. Fix such a time σ_1. Next, look at the time interval when the last quarter of the sinks in $\Xi(i+1)$ are pebbled. Again, the space must go below $3 \cdot M(i)$ pebbles at some point, since otherwise Lemma 7.8 applies. Fix such a time τ_1.

   During the interval $[\sigma_1, \tau_1]$ at least $40 \cdot M(i+1)$ sinks of $\Xi(i+1)$ are pebbled. Let us assume without loss of generality that $20 \cdot M(i+1)$ sinks in the left-hand half are pebbled. Also, there are at most $3 \cdot M(i)$ pebbles on the graph at times σ_1 and τ_1. Since $20 \cdot M(i+1) > 6 \cdot M(i)$, the BLBA-lemma 5.4 applied on $C^{(2)}(i)$ (which we now use for the first time) says that at least

$$N(i) - 6 \cdot M(i) = \bigl(N(0) - 6\bigr) M(i) \geq 80 \cdot M(i) \tag{7.3}$$

   sinks of the Gilbert-Tarjan subgraph $\Xi^{(2)}(i)$ are pebbled during this interval, since at least these many vertices have pebble-free paths to the pebbled sinks of $\Xi(i+1)$.

2. By the discussion above, we can apply the induction hypothesis to $\Xi^{(2)}(i)$ and the interval $[\sigma_1, \tau_1]$. This yields a subinterval $[\sigma_2, \tau_2] \subseteq [\sigma_1, \tau_1]$ such that at least $180 \cdot M(i)$ sources in $\Xi^{(2)}(i)$ are pebbled during this interval and at least $M(i)$ pebbles stays locally on $\Xi^{(2)}(i)$ throughout the whole interval.

3. Consider the subinterval of $[\sigma_2, \tau_2]$ when the first quarter of the sources of $\Xi^{(2)}(i)$ are pebbled. At some point the pebbling space must go below $3 \cdot M(i)$, for otherwise we would be in the special case 2 of Lemma 7.7. Fix such a time σ_3. Looking at the subinterval of $[\sigma_2, \tau_2]$ when the last quarter of the sources in $\Xi^{(2)}(i)$ are pebbled, we again deduce that the space must go below $3 \cdot M(i)$ pebbles at some point if we are not to end up in Lemma 7.7. Fix such a time τ_3.

   During $[\sigma_3, \tau_3]$ at least $90 \cdot M(i)$ sources of $\Xi^{(2)}(i)$ are pebbled, and at least $90 \cdot M(i) - 6 \cdot M(i) \geq 80 \cdot M(i)$ of the sinks in $\Xi^{(1)}(i)$ are connected to these sources by paths of length 1 (a.k.a. edges) that are pebble-free at times σ_3 and τ_3, and must hence also be pebbled.

4. From what has just been said, it follows that we can apply the induction hypothesis to $\Xi^{(1)}(i)$ during $[\sigma_3, \tau_3]$ and obtain a subinterval $[\sigma_4, \tau_4] \subseteq [\sigma_3, \tau_3]$ such that at least $180 \cdot M(i)$ sources in $\Xi^{(1)}(i)$ are pebbled during this interval and at least $M(i)$ pebbles stays locally on $\Xi^{(1)}(i)$ throughout the whole interval.

5. Note that by summing up the pebbles on $\Xi^{(1)}(i)$ and $\Xi^{(2)}(i)$, we now have $M(i) + M(i) = M(i+1)$ pebbles on $\Xi(i+1)$ during the whole interval $[\sigma_4, \tau_4]$, which is the amount required to make the induction step go through. However, we also need to show that a sufficient number of sources in $\Xi(i+1)$ are pebbled during this interval.

   The pattern of reasoning should now look more or less familiar. Consider the subinterval of $[\sigma_4, \tau_4]$ when the first quarter of the sources of $\Xi^{(1)}(i)$ are pebbled. Let σ_5 be some time when the pebbling space drops below $3 \cdot M(i)$, which is guaranteed to happen since Lemma 7.6 would apply otherwise. Similarly, let τ_5 be some point in time during the interval when the last quarter of the sources in $\Xi^{(1)}(i)$ are pebbled at which point the pebbling space is below $3 \cdot M(i)$.

   Consider the left-hand half of the sources in $\Xi(i+1)$. Since at least $90 \cdot M(i)$ sources of $\Xi^{(1)}(i)$ are pebbled during $[\sigma_5, \tau_5]$, and since $90 \cdot M(i) > 6 \cdot M(i)$, the BLBA-lemma applied on $C^{(1)}(i)$ tells us that at least $N(i) - 6 \cdot M(i) = \bigl(N(0) - 6\bigr)M(i) \geq 180 \cdot M(i)$ sources in the left-hand half of $S(\Xi(i+1))$ are pebbled during $[\sigma_5, \tau_5]$. In a completely analogous fashion we deduce that at least $180 \cdot M(i)$ are pebbled in the right-hand half, and summing over the two halves we get the required number of pebbled sources.

We have shown that there is a subinterval $[\sigma_5, \tau_5] \subseteq [\sigma, \tau]$ such that $\min_{t \in [\sigma_5, \tau_5]} \{\mathit{space}(\mathbb{P}_t)\} \geq M(i+1)$ and $\mathcal{P}$ pebbles at least $180 \cdot M(i+1)$ sources in $\Xi(i)$ during $[\sigma_5, \tau_5]$, and the lemma follows. $\square$

Putting all of the pieces together, Lemma 7.4 now follows by the induction principle. The base case is Proposition 7.5. For the induction step, Lemmas 7.6, 7.7, and 7.8 take care of the three special cases, and if none of these cases holds, Lemma 7.9 applies. This concludes the proof.

8 Pebbling Time-Space Trade-offs for Constant Space

In this section we give an exposition of the result by Lengauer and Tarjan [LT82] that even for graphs pebblable in minimal constant space, there are nontrivial time-space trade-offs. More precisely, Lengauer and Tarjan prove the following quadratic trade-offs for constant pebbling space.

Theorem 8.1 ([LT82]). *There are explicitly constructible single-sink DAGs G_n of size $\Theta(n)$ with maximal indegree 2 having the following properties:*

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

It is easy to see that a quadratic time-space trade-off is the strongest possible for pebbling space 3. We conclude the section by briefly discussing whether Theorem 8.1 can be generalized to optimal time-space trade-offs for any constant pebbling space.

[3] The reason we have to include the space constraints in the formal statement of the theorem is that no matter how much space is available, it is of course never possible to do better than linear time.

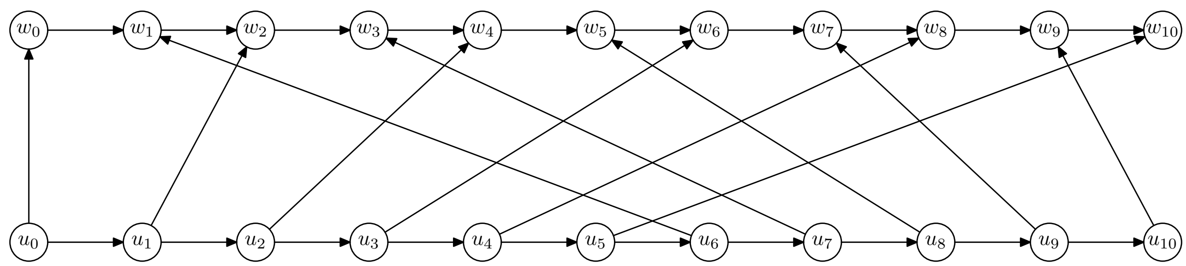


Figure 17: Permutation graph over 11 vertices defined by permutation sending x to $2x \bmod 11$.

8.1 Definition of Permutation Graphs and an Upper Bound

The trade-offs in Theorem 8.1 are obtained for graphs built from permutations in the following way.

Definition 8.2 (Permutation graph ([LT82])). Let π denote some permutation of $\{0, 1, \ldots, n-1\}$. The *permutation graph* $\Delta(n, \pi)$ over n elements with respect to π is defined as follows. $\Delta(n, \pi)$ has $2n$ vertices divided into a *lower row* with vertices $u_0, u_1, \ldots, u_{n-1}$ and an *upper row* with vertices $w_0, w_1, \ldots, w_{n-1}$. For all $i = 0, 1, \ldots, n-2$, there are directed edges (u_i, u_{i+1}) and (w_i, w_{i+1}), and for all $i = 0, 1, \ldots, n-1$, there are edges $(u_i, w_{\pi(i)})$ from the lower row to the upper row.

Thus, the only source vertex in $\Delta(n, \pi)$ is u_0 and the only sink vertex is w_{n-1}. All vertices in the lower row except the leftmost one have indegree 1 and all vertices in the upper row except the leftmost one have indegree 2. Figure 17 shows an example of a permutation graph.

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

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

Lemma 8.3 ([LT82]). *Let $\Delta(n, \pi)$ be the permutation graph over n elements for any permutation π. Then the black pebbling price of $\Delta(n, \pi)$ is $\mathsf{Peb}(\Delta(n, \pi)) = 3$, and for any space s, $3 \leq s \leq n$, there is a black pebbling strategy $\mathcal{P}$ for $\Delta(n, \pi)$ with $\mathsf{space}(\mathcal{P}) \leq s$ and $\mathsf{time}(\mathcal{P}) \leq \frac{2n^2}{s-2} + 2n$.*

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

Proof of Lemma 8.3. The idea is to construct a strategy that pebbles every vertex in the upper row exactly once and uses most of the pebbles to pebble and repebble vertices in the lower row in as time-efficient a manner as possible. To upper-bound the time, we will just count the number of pebble placements and then multiply by 2. We will reserve one pebble for the upper row and $s - 2$ pebbles for the lower row, and keep one auxiliary pebble to use for advancing the other pebbles.

Start by black-pebbling all vertices in $U_1 = \left\{u_{\pi^{-1}(1)}, u_{\pi^{-1}(2)}, \ldots, u_{\pi^{-1}(s-2)}\right\}$ doing at most n pebble placements (and using the auxiliary pebble to get the other pebbles into place). Then use the upper-row pebble and the auxiliary pebble to sweep past $w_1, w_2, \ldots, w_{s-3}$, leaving a black pebble on w_{s-2}.

Now we want to place pebbles on $U_2 = \left\{u_{\pi^{-1}(s-1)}, u_{\pi^{-1}(s)}, \ldots, u_{\pi^{-1}(2(s-2))}\right\}$. In order to save time, we want to use the pebbles already present on the lower row if possible. Let U_1 define a set of intervals $I_1, \ldots, I_{s-1}$ covering the lower row such that the leftmost interval I_1 contains no pebbles from U_1 (and is possibly empty) and all other intervals $I_2, \ldots, I_{s-1}$ contains a pebble from U_1 on their leftmost vertex

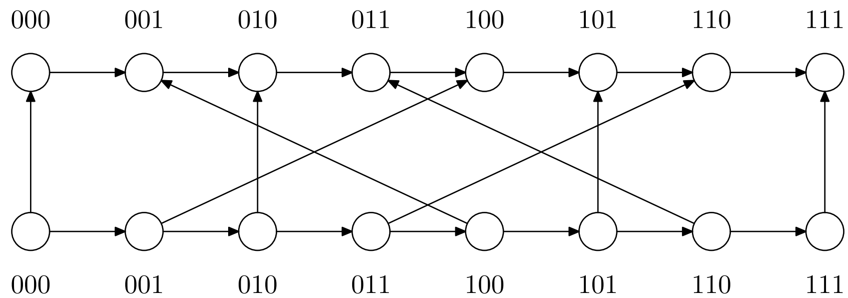


Figure 18: Bit reversal graph $\Delta(8, \mathrm{rev})$ on 8 elements.

and no other pebbles. We can now pebble U_2 from left to right in the following way. If $U_2 \cap I_1 \neq \emptyset$, there is some other interval I_j such that $U_2 \cap I_j$ is empty. Use the pebble in U_1 from this interval and the auxiliary pebble to sweep a black pebble from u_0 to the leftmost vertex $u^* \in U_2 \cap I_1$. If $\left|U_2 \cap I_1\right| > 1$, there must exist for every vertex in this set another empty interval $I_{j'}$ and we use this pebble and the auxiliary pebble to advance a black pebble from u^* onwards to this vertex. Continuing in this fashion for all intervals $I_2, \ldots, I_{s-1}$, we place black pebbles on all vertices in U_2, and then use the upper-row and auxiliary pebbles to pebble and unpebble $w_{s-1}, w_s, \ldots, w_{2s-5}$, leaving a black pebble on w_{2s-4}. We progress in the same way for the upper-row vertices $w_{2(s-2)+1}, \ldots, w_{3(s-2)}$ by first placing black pebbles on $U_3 = \left\{u_{\pi^{-1}(2(s-2)+1)}, u_{\pi^{-1}(s)}, \ldots, u_{\pi^{-1}(3(s-2))}\right\}$ in the lower row in a completely analogous manner to what was done for U_2 above, and continue until finally w_{n-1} is pebbled.

This pebbling strategy has $\lceil n/(s-2) \rceil$ phases corresponding to the U_i:s, and in every phase at most $n-(s-2)$ pebble placements are made on the lower row, except for the first phase when up to n placements can be made. Each vertex in the upper row is pebbled exactly once. All in all, the number of pebble placements is at most

$$(n-(s-2))\left(\left\lceil \frac{n}{s-2} \right\rceil - 1\right) + 2n \leq \frac{n^2}{s-2} + n \ , \tag{8.1}$$

and adjusting for pebble removals by multiplying by 2 we get the time bound stated in the lemma. □

8.2 Bit Reversal Graphs and Lower Bounds

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

Definition 8.4 (Bit reversal graph ([LT82])). The *m-bit reversal* of i, $0 \leq i \leq 2^m - 1$, is the integer $\mathrm{rev}_m(i)$ obtained by writing the m-bit binary representation of i in reverse order. The *bit reversal graph* $\Delta(2^m, \mathrm{rev}_m)$ is the permutation graph over $n = 2^m$ with respect to rev_m.

For instance, we have $\mathrm{rev}_3(1) = 4$, $\mathrm{rev}_3(2) = 2$, and $\mathrm{rev}_3(3) = 6$. We will denote the bit reversal graph by $\Delta(n, \mathrm{rev})$ for simplicity, implicitly assuming that $n = 2^m$. An example of a bit reversal graph can be found in Figure 18.

For bit reversal graphs, the trade-off in Lemma 8.3 for black pebbling is asymptotically tight.

Theorem 8.5 ([LT82]). *Suppose that $\mathcal{P}$ is any complete black pebbling of the bit reversal graph $\Delta(n, \mathrm{rev})$ over $n = 2^m$ elements such that $\mathsf{space}(\mathcal{P}) = s$ for $s \geq 3$. Then $\mathsf{time}(\mathcal{P}) \geq \frac{n^2}{8s}$.*

Proof. Let us assume that $s \leq n/4$, since otherwise the theorem is trivially true. Let r be the integer such that $2s \leq 2^r < 4s$. Divide the upper row into 2^{n-r} intervals

$$I_j = \left\{w_{j \cdot 2^r},\ w_{j \cdot 2^r + 1},\ \ldots,\ w_{(j+1) \cdot 2^r - 1}\right\} \tag{8.2}$$

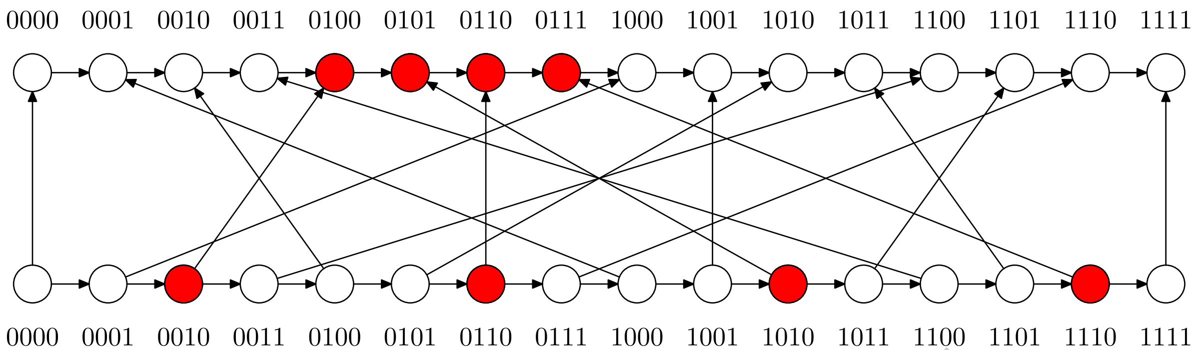


Figure 19: Upper-row vertices $w_{j\cdot 2^r}, w_{j\cdot 2^r+1}, \ldots, w_{(j+1)\cdot 2^r-1}$ split lower row into evenly sized intervals.

of length 2^r for $0 \le j < 2^{m-r}$. Let τ_j be the first time $w_{(j+1)\cdot 2^r-1}$ is pebbled, and define $\tau_{-1} = 0$. Clearly, we must have $\tau_j > \tau_{j-1}$ in any black-only pebbling. We want to lower-bound $\tau_j - \tau_{j-1}$.

At time τ_{j-1}, all vertices in I_j are pebble-free, so all of these 2^r vertices are pebbled during $[\tau_{j-1}, \tau_j]$. Now look at the set of vertices

$$\mathrm{rev}_m^{-1}(I_j) = \left\{u_i \,\middle|\, i = \mathrm{rev}_m^{-1}\big(j\cdot 2^r\big), \mathrm{rev}_m^{-1}\big(j\cdot 2^r+1\big), \ldots, \mathrm{rev}_m^{-1}\big((j+1)\cdot 2^r - 1\big)\right\} \tag{8.3}$$

in the lower row. (Figure 19 illustrates $I_1 = \{w_4, w_5, w_6, w_7\}$ and $\mathrm{rev}_m^{-1}(I_1)$ for $r = 2$ in the bit reversal DAG over 16 elements.) By the definition of bit reversal permutations, every I_j divides the lower row into $2^r - 1$ intervals of length exactly 2^{m-r}. To see this, note that rev_m^{-1} fixes the $n - r$ lower bits to the bit pattern $j \cdot 2^r$ reversed, while the r upper bits run through all combinations of 0 and 1. Disregarding the leftmost and rightmost intervals, we get $2^r - 1$ intervals of length exactly 2^{m-r} in between them.

At time τ_{j-1}, there are at most $s - 2$ pebbles on the lower row, so at least $2^r - 1 - (s-2) > s$ of the intervals defined by $\mathrm{rev}_m^{-1}(I_j)$ are completely pebble-free at this time. When we reach time τ_j, all of these intervals must have been completely pebbled. This requires strictly more than $s \cdot 2^{m-r} > s \cdot \frac{n}{4s} = n/4$ pebble placements between times τ_{j-1} and τ_j. Multiplying by 2 to account for pebble removals as well, and summing over all intervals $[\tau_{j-1}, \tau_j]$, we get that the total time of the pebbling is at least

$$\sum_{j=0}^{2^{m-r}-1} \big(\tau_j - \tau_{j-1}\big) \;>\; 2^{m-r} \cdot \frac{n}{2} \;>\; \frac{n^2}{8s} \tag{8.4}$$

and the theorem follows. ☐

Note, in particular, that if we want to black-pebble $\Delta(n, \mathrm{rev})$ in linear time, then linear space is needed. If we are also allowed to use white pebbles, however, the argument in the proof of Theorem 8.5 breaks down since we can no longer assume that the pebbling proceeds through the DAG in topological order. Modifying the argument to take into account the possibility that intervals are pebbled in arbitrary order, we get the following lower bound.

Theorem 8.6 ([LT82]). *Let $\mathcal{P}$ be any complete black-white pebbling of $\Delta(n, \mathrm{rev})$ with $\mathsf{space}(\mathcal{P}) = s$ for $s \ge 3$. Then $\mathsf{time}(\mathcal{P}) \ge \frac{n^2}{18s^2} + 2n$.*

Proof. Suppose that $s < n/6$ since otherwise the statement is trivially true. Write $m = \log n$ and fix r such that $3s \le 2^r < 6s$. Divide the vertices in the upper row into $2^{m-r} > n/6s$ intervals I_j, $0 \le j < 2^{m-r}$, as in (8.2). Let $\tau_0 = 0$ and $M_0 = \emptyset$, and inductively define τ_i to be the first time after τ_{i-1} when the first interval $I_j \notin M_{i-1}$ has been pebbled and unpebbled completely. At time τ_i, a pebble is removed from I_j

and at most $s-1$ other intervals $I_{j'}$ contain pebbles. Let M_i be the union of M_{i-1} and the at most s intervals just mentioned, including I_j. Repeat this procedure until M_i covers all intervals (which clearly must be the case at the end of the pebbling).

There are strictly more than $n/6s$ intervals, and at most s new intervals are added to M_i at each iteration. Hence, the above procedure is repeated at least $\lceil n/6s^2 \rceil$ times. We claim that in between τ_{i-1} and τ_i, there are at least $n/6$ pebble placements made on the lower row. To see this, note first that by construction I_j is empty at time τ_{i-1}, so all of I_j is pebbled during $[\tau_{i-1}, \tau_i]$. The immediate predecessors of the vertices in I_j in the lower row again divides this row into $2^r - 1$ intervals of length 2^{m-r} (plus the smaller-sized leftmost and rightmost intervals, which we ignore). At time τ_{i-1}, at most $s-1$ of these intervals in the lower row contain pebbles, and at time τ_i at most $s-1$ other intervals contain pebbles. By Lemma 3.11 on page 13, all the other at least $2^r - 2(s-1) > s$ intervals in the lower row must be completely pebbled and unpebbled during $[\tau_{i-1}, \tau_i]$. But this requires more than $s \cdot 2^{m-r} > s \cdot n/6s = n/6$ pebble placements.

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

The reason for the discrepancy between Theorem 8.5 and Theorem 8.6 turns out to be that in fact, it *is* possible to do better using white pebbles in addition to the black ones. In particular, there is a linear-time black-white pebbling strategy for $\Delta(n, \mathrm{rev})$ using only order of $\sqrt{n}$ pebbles.

Theorem 8.7 ([LT82]). *For any space $s \geq 3$ there is a complete black-white pebbling $\mathcal{P}$ of $\Delta(n, \mathrm{rev})$ with $\mathsf{space}(\mathcal{P}) \leq s$ and $\mathsf{time}(\mathcal{P}) \leq 144\frac{n^2}{s^2} + 12n$.*

The main work in proving the theorem is the next lemma. We establish the lemma first and then explain how it implies Theorem 8.7.

Lemma 8.8 ([LT82]). *For all s, $3 \leq s \leq 3\sqrt{n}$, there is a complete pebbling of $\Delta(n, \mathrm{rev})$ in space at most s and time at most $144\frac{n^2}{s^2} + 2n$.*

Proof of Lemma 8.8. Write $m = \log n$ and let r be the non-negative integer such that

$$3 \cdot 2^r \leq s < 3 \cdot 2^{r+1} \quad . \tag{8.5}$$

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

$$I_j = \left\{ w_{j \cdot 2^{m-r}+k} \,\middle|\, k = 0, 1, \ldots, 2^{m-r} - 1 \right\} \tag{8.6}$$

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

$$C_j^i = \left\{ w_{j \cdot 2^{m-r} + i \cdot 2^r + k} \,\middle|\, k = 0, 1, \ldots, 2^r - 1 \right\} \tag{8.7}$$

for $i = 0, \ldots, 2^{m-2r} - 1$. Note that we must have $2^{m-2r} \geq 1$ for this definition to make sense, but this holds since $s \leq 3\sqrt{n}$ by assumption. Figure 20 exemplifies these definitions on the 32-element bit reversal DAG with 2^2 intervals and 2 chunks per interval.

The pebbling strategy will proceed in 2^{m-2r} *phases* corresponding to the 2^{m-2r} chunks in each interval, and in 2^r *stages* within each phase corresponding to the different intervals. All the phases in the pebbling are completely analogous except for some minor tweaks in the first and final phases, which we refer to as the 0th and $(2^{m-2r} - 1)$st phases, respectively. To help the reader parse the notation, we note that in what follows superscripts i will correspond to phases/chunks and subscripts j to stages/intervals. We reserve 2^r pebbles for the lower row, 2^r pebbles for the "current chunks" in the upper row, and $2^r - 1$ pebbles for the

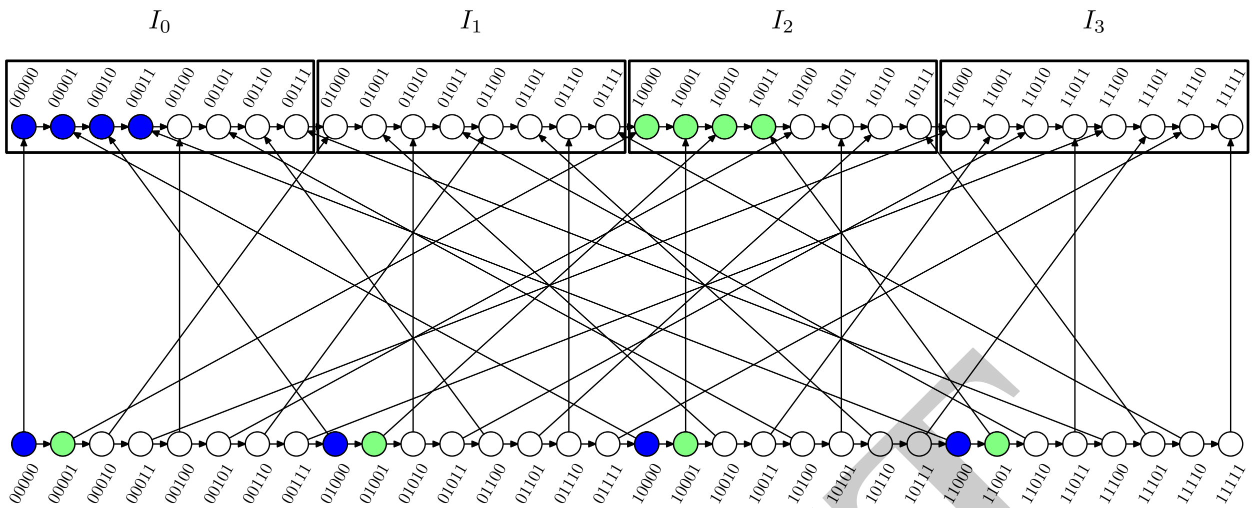


Figure 20: Intervals I_j for $r = 2$ in $\Delta(32, \mathrm{rev})$ and 0th chunks in I_0 and $I_{\mathrm{rev}_r(1)} = I_2$ with inverse images.

rightmost vertices in $I_0, I_1, \ldots, I_{2^r-2}$. By (8.5), this leaves one auxiliary pebble to help with advancing the other pebbles.

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

$$U_0^0 = \{u_{\mathrm{rev}_m(k)} \mid k = 0, 1, \ldots, 2^r - 1\} \tag{8.8}$$

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

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

$$U_1^0 = \left\{u_{\mathrm{rev}_m(\mathrm{rev}_r(1) \cdot 2^{m-r}+k)} \,\middle|\, k = 0, 1, \ldots, 2^r - 1\right\} \;, \tag{8.9}$$

which by (8.7) is the set of all predecessors in the lower row of the 0th chunk $C^0_{\mathrm{rev}_r(1)}$ of the interval $I_{\mathrm{rev}_r(1)}$ (see Figure 20 for a concrete example of this). If we place a white pebble on the rightmost vertex of the interval $I_{\mathrm{rev}_r(1)-1}$, this white pebble plus the lower-row black pebbles on U_1^0 allow us to advance a black pebble along all the vertices of the 0th chunk of $I_{\mathrm{rev}_r(1)}$, leaving it on the rightmost vertex. This concludes stage 1 of phase 0.

Continuing in this way, in the jth stage of phase 0 we can move the lower-row pebbles from U_{j-1}^0 to U_j^0 where this notation is generalized to mean

$$U_j^0 = \left\{u_{\mathrm{rev}_m(\mathrm{rev}_r(j) \cdot 2^{m-r}+k)} \,\middle|\, k = 0, 1, \ldots, 2^r - 1\right\} \tag{8.10}$$

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

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

$$U_j^i = \left\{ u_{\mathrm{rev}_m(\mathrm{rev}_r(j)\cdot 2^{m-r} + i\cdot 2^r + k)} \;\middle|\; k = 0, 1, \ldots, 2^r - 1 \right\} = \mathrm{rev}_m^{-1}\left(C^i_{\mathrm{rev}_r(j)} \right) \quad , \tag{8.11}$$

where the second equality is easily verified from (8.7). In stage 0 of phase i, we rearrange the lower-row black pebbles so that they cover the vertices in U_0^i. Since the 2^r black pebbles are already present somewhere on the lower row, this can be achieved with at most $n - 2^r$ pebble placements (as in the proof of Lemma 8.3). This allows us to advance the pebble in I_0 on the upper row from the rightmost vertex in chunk $i-1$ to the rightmost vertex in chunk i. Moving the pebbles in U_0^i one step to the right in each following stage to U_1^i, U_2^i, et cetera, we can sweep black pebbles across the ith chunks of the other intervals I_j in the order $I_{\mathrm{rev}_r(1)}, I_{\mathrm{rev}_r(2)}, \ldots, I_{\mathrm{rev}_r(2^r-1)} = I_{2^r-1}$. All in all, we make at most $(n - 2^r) + (2^r - 1)\cdot 2^r$ pebble placements on the lower row during phase i for $i \geq 1$.

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

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

$$\begin{aligned} 2^{m-2r}\big((n - 2^r) + (2^r - 1)\cdot 2^r\big) + 2^r + 2n &< 2^{m-2r}\big(n + 2^{2r}\big) + 2n \\ &\leq 72\frac{n^2}{s^2} + 2n \end{aligned} \tag{8.12}$$

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

Proof of Theorem 8.7. For $s \leq 3\sqrt{n}$ this is the same statement as in Lemma 8.8 (and note that for $s < 70$, the black-only pebbling in Lemma 8.3 gives a better time bound). To get the statement for $s > 3\sqrt{n}$, use the same pebbling strategy as in the proof of Lemma 8.8 but choose r so that $\sqrt{n}/2 < 2^r \leq \sqrt{n}$. Then the number of chunks 2^{m-2r} is at most 2, and the time bound derived from (8.12) reduces to $12n$. □

On a high level, the reason that black-white pebblings can do much better than black-only pebblings on bit reversal DAGs is that these graphs have such a regular structure. Lengauer and Tarjan raise the question whether there are other permutations for which the lower bound in Theorem 8.5 holds also for black-white pebbling.

Open Problem 4 ([LT82]). *Are there families of permutations π_n over n elements such that for any black-white pebbling strategies $\mathcal{P}_n$ for the permutation graphs $\Delta(n, \pi_n)$, it must hold that $\mathsf{time}(\mathcal{P}_n) = \Omega\big(n^2/\mathsf{space}(\mathcal{P}_n)\big)$?*

Lengauer and Tarjan conjecture that the answer to this question is yes, but to the best of our knowledge, the problem has remained open. One could ask whether anything interesting can be said about what holds for a random permutation in this respect. If the conjecture turns out to be true for a random permutation (with high probability, say), then such a result, although non-constructive, would be interesting.

8.3 Optimal Time-Space Trade-offs for Any Constant Space?

As we mentioned at the start of this section, it is easy to see that any (black or black-white) pebbling in constant space s can take time at most $\mathrm{O}\big(n^{s-1}\big)$. Let us write this down as a formal observation.

Observation 8.9. *If G is a single-sink DAG on n vertices that can be pebbled (by a black-white or black-only pebbling) in constant space s, then there is a pebbling strategy for G (black-white or black-only, respectively) in space s and time $\mathrm{O}\big(n^{s-1}\big)$.*

Proof. As was noted in the introduction, any pebbling in space s of a graph with n vertices need not take more time than $2\sum_{r=0}^{s} 2^r\binom{n}{r} = \mathrm{O}\big(s2^s \cdot n^s\big) = \mathrm{O}\big(n^s\big)$ (since s is a constant). This is so since each possible distinct pebble configuration can only appear and disappear once during a non-redundant pebbling. However, whenever we have a configuration with s pebbles, the next move will be an erasure bringing us down to a configuration with $s-1$ pebbles. Thus, we only need to count the number of distinct configurations with at most $s-1$ pebbles, which gives the time bound $\mathrm{O}\big(n^{s-1}\big)$. ☐

One remarkable aspect of Theorem 8.1 is that the trade-offs in that theorem meet this almost trivial upper bound established by simple counting for pebbling space $s = 3$. It is a natural question whether similar trade-offs can be proven for any constant space s.

Open Problem 5. *Is it possible to prove trade-offs on the form in Theorem 8.1 with pebbling time $\mathrm{O}\big(n^{s-1}\big)$ and pebbling space s for any constant space s?*

One simple first idea to try would be to study "multi-layered" versions of the bit reversal DAGs in Definition 8.4 with $s-1$ layers of n vertices each, and with edges between consecutive layers according to the bit reversal permutation (so that any two consecutive layers form a copy of the bit reversal DAG).

9 Pebbling Trade-offs for Arbitrarily Small Non-constant Space

It is clear that we can never get superpolynomial trade-offs from DAGs pebblable in constant space, since such graphs must have constant-space pebbling strategies in polynomial time by Observation 8.9. However, perhaps somewhat surprisingly, as soon as we study any *arbitrarily slowly* growing function, we can obtain superpolynomial black and black-white pebbling trade-offs for graph families with pebbling price growing as slowly as this function.

Theorem 9.1 ([Nor10a]). *Let $g(n)$ be any arbitrarily slowly growing monotone function $\omega(1) = g(n) = \mathrm{O}\big(n^{1/7}\big)$, and let $\epsilon > 0$ be an arbitrarily small positive constant. Then there is a family of explicitly constructible single-sink DAGs $\{G_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

1. *The graph G_n has black-white pebbling price $\mathsf{BW\text{-}Peb}^{\bullet}(G) = g(n) + \mathrm{O}(1)$ and black pebbling price $\mathsf{Peb}^{\bullet}(G) = 2 \cdot g(n) + \mathrm{O}(1)$.*
2. *There is a complete black pebbling $\mathcal{P}$ of G_n with $\mathsf{time}(\mathcal{P}) = \mathrm{O}(n)$ and $\mathsf{space}(\mathcal{P}) = \mathrm{O}\left(\sqrt[3]{n/g^2(n)}\right)$*
3. *Any complete black-white pebbling $\mathcal{P}$ of G_n in space at most $\big(n/g^2(n)\big)^{1/3-\epsilon}$ requires pebbling time superpolynomial in n.*

We remark that the upper-bound condition $g(n) = \mathrm{O}\big(n^{1/7}\big)$ is very mild and is there only for technical reasons in this theorem. If we allow the minimal space to grow as fast as n^{ϵ} for some $\epsilon > 0$, then there are other pebbling trade-off results that can give even stronger results for resolution than the one stated above (see, for instance, Section 11). Thus the interesting part is that $g(n)$ is allowed to grow arbitrarily slowly.

Theorem 9.1 follows from proving strong trade-off properties for a very simple but surprisingly versatile family of graphs, which we describe next.

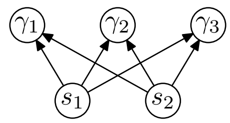


Figure 21: Base case $\Gamma(3,1)$ for Carlson-Savage graph with 3 spines and sinks.

9.1 Definition of Graph Family and Statement of Results

Our graph family is built on a construction by Carlson and Savage [CS80, CS82]. Carlson and Savage only prove their trade-off for black pebbling, however, and in order to get results for black-white pebbling we have to modify the construction somewhat and also apply some new ideas in the proofs. The next definition will hopefully be easier to parse if the reader first studies the illustrations in Figures 21 and 22.

Definition 9.2 (Carlson-Savage graph [CS80, CS82, Nor10a]). The two-parameter graph family $\Gamma(c,r)$, for $c,r \in \mathbb{N}^+$, is defined by induction over r. The base case $\Gamma(c,1)$ is a DAG consisting of two sources s_1, s_2 and c sinks $\gamma_1,\ldots,\gamma_c$ with directed edges (s_i,γ_j), for $i = 1,2$ and $j = 1,\ldots,c$, i.e., edges from both sources to all sinks. The graph $\Gamma(c,r+1)$ has c sinks and is built from the following components:

- c disjoint copies $\Pi^{(1)}_{2r},\ldots,\Pi^{(c)}_{2r}$ of a pyramid (Definition 4.5) of height $2r$, where we let $z_1,\ldots,z_c$ denote the pyramid sinks.
- one copy of $\Gamma(c,r)$, for which we denote the sinks by $\gamma_1,\ldots,\gamma_c$.
- c disjoint and identical *spines*, where each spine is composed of cr *sections*, and every section contains $2c$ vertices. We let the vertices in the ith section of a spine be denoted $v[i]_1,\ldots,v[i]_{2c}$.

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

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

The graph family $\Gamma^*(c,r)$ is defined analogously except that $\Gamma^*(c,r+1)$ contains pyramids $\Pi^{(1)}_r,\ldots,\Pi^{(c)}_r$ of height r and each of the c spines contain only r sections.

We now make the formal statements of the trade-off properties that these DAGs possess. The proofs of allt the statements are postponed to Section 9.2. First, we collect some basic properties.

Lemma 9.3 ([CS82, Nor10a]). *The graphs $\Gamma(c,r)$ are of size $\bigl|V\bigl(\Gamma(c,r)\bigr)\bigr| = \Theta\bigl(cr^3 + c^3r^2\bigr)$, and have black-white pebbling price $\mathit{BW\text{-}Peb}^{\emptyset}\bigl(\Gamma(c,r)\bigr) = r+2$ and black pebbling price $\mathit{Peb}^{\emptyset}\bigl(\Gamma(c,r)\bigr) = 2r+1$.*

The graphs $\Gamma^*(c,r)$ are of size $\Theta\bigl(cr^3 + c^2r^2\bigr)$ and have black pebbling price $\mathit{Peb}^{\emptyset}\bigl(\Gamma^*(c,r)\bigr) = r+2$.

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

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

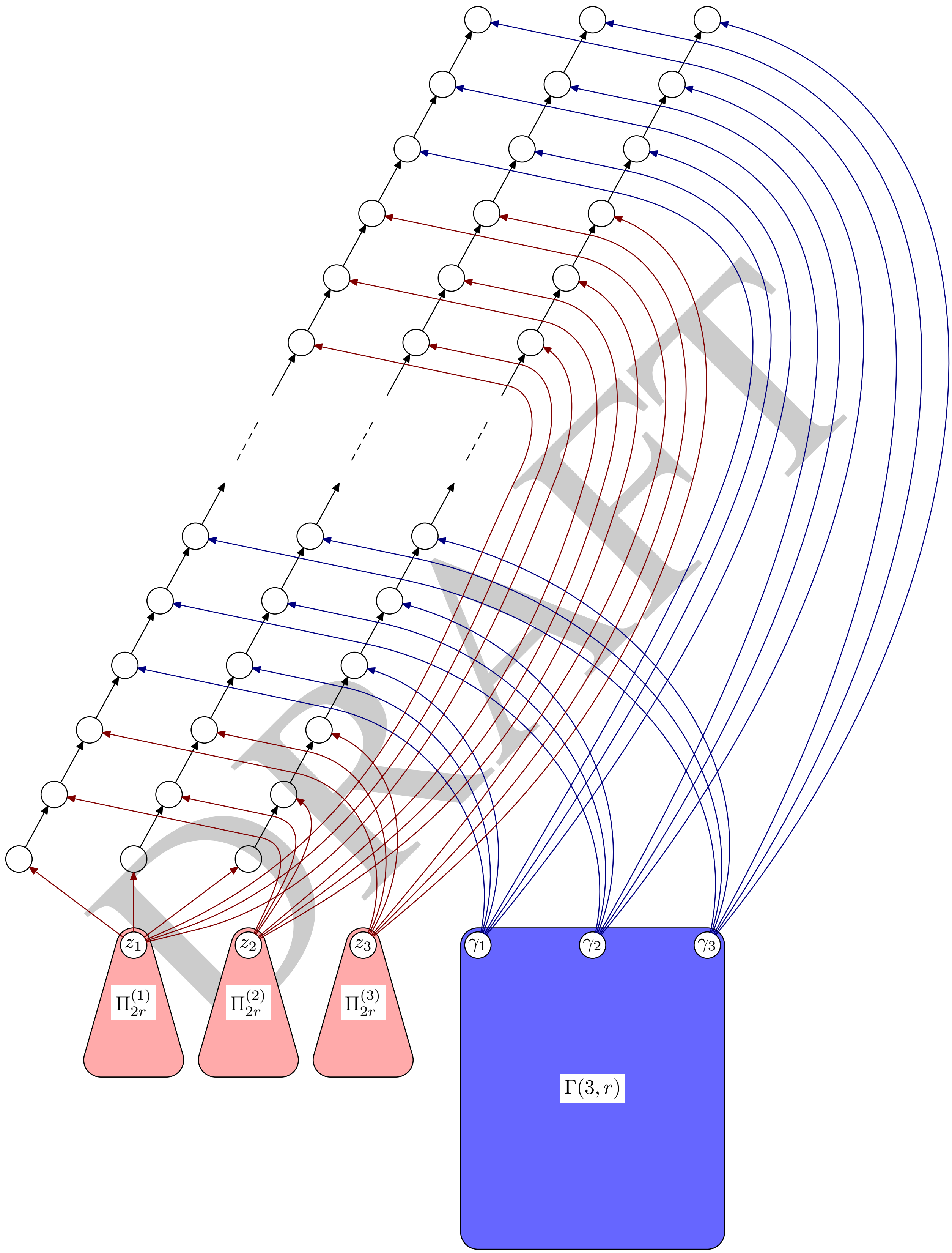


Figure 22: Inductive definition of Carlson-Savage graph $\Gamma(3, r+1)$ with 3 spines and sinks.

Lemma 9.4 ([Nor10a]). *The graphs $\Gamma(c,r)$ and $\Gamma^*(c,r)$ have persistent black pebbling strategies in simultaneous space $\mathrm{O}(c+r)$ and time linear in the size of the graphs.*

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

Carlson and Savage prove the following black pebbling trade-off.

Theorem 9.5 ([CS82]). *Suppose that $\mathcal{P}$ is a complete visiting black pebbling of $\Gamma^*(c,r)$ in space less than $\mathsf{Peb}^{\emptyset}\big(\Gamma^*(c,r)\big)+s=(r+2)+s$ for $0<s\le c-3$. Then*

$$\mathsf{time}(\mathcal{P})\ge\left(\frac{c-s}{s+1}\right)^{r}\cdot r!\ .$$

The main result of this section is an extension of Theorem 9.5 to black-white pebbling, which will allow us to get a variety of pebbling trade-off results if we choose the parameters c and r appropriately.

Theorem 9.6 ([Nor10a]). *Suppose that $\mathcal{P}$ is a complete visiting black-white pebbling of $\Gamma(c,r)$ with*

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

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

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

Before proving the lemmas and theorems above, let us see how they yield Theorem 9.1.

Proof of Theorem 9.1. Consider the graphs $\Gamma(c,r)$ in Definition 9.2. We want to choose the parameters c and r in a suitable way so that get a family of graphs in size $n=\Theta\big(cr^3+c^3r^2\big)$ (using the bound on the size of $\Gamma(c,r)$ from Lemma 9.3). If we choose $r=r(n)=g(n)$ for $g(n)=\mathrm{O}\big(n^{1/7}\big)$, this forces $c=c(n)=\Theta\big(\sqrt[3]{n/g^2(n)}\big)$. Consider the graph family $\{H_n\}_{n=1}^{\infty}$ defined by $H_n=\Gamma(c(n),r(n))$ as above and let $G_n=\widehat{H_n}$ be the single-sink version of H_n. This is a family of single-sink DAGs of size $\Theta(n)$.

By Lemma 9.3 and Observation 3.8 it holds that $\mathsf{Peb}(G_n)=g(n)+\mathrm{O}(1)$. Also, the persistent black pebbling of H_n in Lemma 9.4 yields a linear-time pebbling of G_n in space $\mathrm{O}\big(\sqrt[3]{n/g^2(n)}\big)$.

Now set the parameter s in Theorem 9.6 to $s=c^{1-\epsilon'}$ for $\epsilon'=3\epsilon$. Then for large enough n we have $s\le c/8-1$ and Theorem 9.6 can be applied. We get that if the pebbling space is less than $\big(n/g^2(n)\big)^{1/3-\epsilon}$, then the required time for the black-white pebbling grows as $\big(\Omega\big(c^{\epsilon'}\big)\big)^{r}=\big(\Omega\big(n/g^2(n)\big)\big)^{\epsilon g(n)}$ which is superpolynomial in n for any $g(n)=\omega(1)$. The theorem follows. ☐

All proofs will be presented in Section 9.2, but let us try to provide some intuition as to why the theorem should be true. For simplicity, let us focus on the black-only pebbling case in Theorem 9.5. Inductively, suppose that the trade-off in Theorem 9.5 has been proven for $\Gamma^*(c,r)$ and consider $\Gamma^*(c,r+1)$. Any pebbling strategy for this DAG will have to pebble through all sections in all spines. Consider the first section anywhere, let us say on spine j, that has been completely pebbled, i.e., there have been pebbles placed on and removed from all vertices in the section. Let us say that this happens at time τ_1. But this means that $\Gamma^*(c,r)$ and all pyramids $\Pi^{(1)}_{r},\ldots,\Pi^{(c)}_{r}$ must have been completely pebbled during this part of the pebbling as well. Fix any pyramid and consider some point in time $\sigma_1<\tau_1$ when the number of pebbles in this pyramid reaches the space $r+2$ required by the known lower bound on pyramid pebbling price. At this point, the rest of the graph must contain very few pebbles. In particular, there are very few pebbles on the subgraph $\Gamma^*(c,r)$ at time σ_1, so for all practical purposes we can think of $\Gamma^*(c,r)$ as being essentially empty of pebbles.

Let us now shift the focus to the next section in the spine j that is completed, say, at time $\tau_2 > \tau_1$. Again, we can argue that some pyramid is completely pebbled in the time interval $[\tau_1, \tau_2]$, and thus has $r+2$ pebbles on it at some time $\sigma_2 > \tau_1 > \sigma_1$. This means that we can think of $\Gamma^*(c,r)$ as being essentially empty at time σ_2 as well.

But note that all sinks in the subgraph $\Gamma^*(c,r)$ must have been pebbled in the time interval $[\sigma_1, \sigma_2]$, and since we know that $\Gamma^*(c,r)$ is (almost) empty at times σ_1 and σ_2, this allows us to apply the induction hypothesis. Since $\mathcal{P}$ has to pebble through a lot of sections in different spines, we will be able to repeat the above argument many times and apply the induction hypothesis on $\Gamma^*(c,r)$ in each round. Adding up all the lower bounds obtained in this way, the induction step goes through.

This is the spirit of the proof of Theorem 9.5 in [CS82], although there are of course a number of technical details to take care of. For black-white pebbling, things are more complicated. The main problem is that in contrast to a black pebbling, that has to proceed through the DAG in some kind of bottom-up fashion, a black-white pebbling can place and remove pebbles anywhere in the DAG at any time. Therefore, it is more difficult to control the progress of a black-white pebbling, and one has to work harder in the proof.

Indeed, it should be noted that the added complications when going from black to black-white pebbling result in our bound for black-white pebbling being slightly worse than the one in [CS82] for black pebbling only. More specifically, Carlson and Savage are able to prove their results for the DAGs $\Gamma^*(c,r)$ having only $\Theta(r)$ sections per spine, whereas we need $\Theta(cr)$ sections in $\Gamma(c,r)$. This blows up the number of vertices, which in turn weakens the trade-offs measured in terms of graph size. It would be interesting to find out whether our proof, soon to be presented in Section 9.2, could in fact be made to work for graphs with only $\mathrm{O}(r)$ sections per spine. If so, this would immediately improve all the trade-off result Theorem 9.1 as well as the other results in this survey that we obtain from the Carlson-Savage graph family.

Open Problem 6 ([Nor10a]). *Is it possible to prove a black-white pebbling trade-off as in Theorem 9.6 for the DAGs $\Gamma^*(c,r)$, or for a modification of $\Gamma^*(c,r)$ having only $\Theta(r)$ sections per spine? Or are $\Omega(cr)$ sections in fact needed to get a trade-off for black-white pebbling?*

9.2 Proofs of the Carlson-Savage Graph Properties

Before proving the results claimed in Section 9.1, we show a couple of useful auxiliary lemmas. The first lemma below gives us information about how the spines in the Carlson-Savage DAGs are pebbled. We will use this information repeatedly in what follows.

Lemma 9.7. *Suppose that G is a DAG and that v is a vertex in G with a path Q to some sink $z_i \in Z(G)$ such that all vertices in $Q \setminus \{z_i\}$ have outdegree 1. Then any frugal black-white pebbling strategy pebbles v exactly once, and the path Q contains pebbles during one contiguous time interval.*

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

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

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

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

Lemma 9.8. *Let G be a DAG and H a subgraph in G such that H has a unique sink z_h and the only edges between $V(H)$ and $V(G) \setminus V(H)$ emanate from z_h. Suppose that $\mathcal{P}$ is any frugal complete pebbling of G having the property that H is completely empty of pebbles at some given time τ' but at least one vertex of H has been pebbled during the time interval $[0, \tau']$. Then $\mathcal{P}$ pebbles H completely during the interval $[0, \tau']$.*

Proof. Suppose that $v \in V(H)$ is pebbled at time $\sigma' < \tau'$. Note that all paths starting in v must hit z_h sooner or later, since z_h is the unique sink of H and there is no other way out of H into the rest of G. Consider the longest path from v to z_h. If this path has length 1, clearly z_h must be pebbled before time τ' since otherwise the pebble placement on v is non-essential. The same statement follows for any v by induction over the path length. But since H is empty at times 0 and τ' and z_h is pebbled during $(0, \tau')$, H is completely pebbled during this time interval. ☐

Let us now establish that the size and pebbling price of the Carlson-Savage DAGs are as claimed.

Proof of Lemma 9.3. We only present the proofs for $\Gamma(c, r)$. The claims for $\Gamma^*(c, r)$ are proven in a completely analogous fashion.

The base case graph $\Gamma(c, 1)$ in Definition 9.2 has size $c+2$. A pyramid of height h has $(h+1)(h+2)/2$ vertices, so the c pyramids of height $2(r-1)$ in $\Gamma(c, r)$ contribute $cr(2r-1)$ vertices. The c spines with cr sections of $2c$ vertices each contribute a total of $2c^3 r$ vertices. And then there are the vertices in $\Gamma(c, r-1)$. Summing up, the total number of vertices in $\Gamma(c, r)$ is

$$(c+2) + \sum_{i=2}^{r} \bigl(ci(2i-1) + 2c^3 i\bigr) = \Theta\bigl(cr^3 + c^3 r^2\bigr) \tag{9.1}$$

as is stated in the lemma.

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

Suppose inductively that $\mathit{BW\text{-}Peb}^{\emptyset}(\Gamma(c, r)) = r + 2$ and consider $\Gamma(c, r+1)$. It is straightforward to see that $\mathit{BW\text{-}Peb}^{\emptyset}(\Gamma(c, r+1)) \leq r+3$. Every pyramid $\Pi_{2r}^{(j)}$ can be completely pebbled with $r+2$ pebbles (Theorem 4.2). We can pebble each spine bottom-up in the following way, section by section. Suppose by induction that we have a black pebble on the last vertex $v[i-1]_{2c}$ in the $(i-1)$st section. Keeping the pebble on $v[i-1]_{2c}$, perform a complete visiting pebbling of $\Pi_{2r}^{(1)}$. At some point during this pebbling we must have a pebble on the pyramid sink z_1 and at most r other pebbles on the pyramid (by Proposition 3.9). At this time, place a black pebble on $v[i]_1$ and remove the pebble from $v[i-1]_{2c}$. Complete the pebbling of $\Pi_{2r}^{(1)}$, leaving the pyramid empty. Performing complete visiting pebblings of $\Pi_{2r}^{(2)}, \ldots, \Pi_{2r}^{(c)}$ in an analogous fashion allows us to move the black pebble along $v[i]_2, \ldots, v[i]_c$, never exceeding total pebbling space $r+3$. In the same way, for every visiting pebbling $\mathcal{P}$ of $\Gamma(c, r)$ there must exist times σ_i for all $i = 1, \ldots, c$, when $\mathit{space}(\mathbb{P}_{\sigma_i}) < \mathit{space}(\mathcal{P})$ and the sink γ_i contains a pebble. Performing a minimum-space pebbling of $\Gamma(c, r)$, possibly c times if necessary, this allows us to advance the black pebble along $v[i]_{c+1}, \ldots, v[i]_{2c}$, never exceeding total pebbling space $r+3$. This shows that $\Gamma(c, r+1)$ can be completely pebbled with $r+3$ pebbles. A simple syntactic adaptation of this arguments for black pebbling (appealing to Theorem 4.2 for the black pebbling price of pyramids) also yields $\mathit{Peb}^{\emptyset}(\Gamma(c, r)) \leq 2r + 3$.

To prove that there are matching lower bounds for the pebbling constructed above, it is sufficient to show that some pyramid $\Pi_{2r}^{(j)}$ must be completely pebbled while there is at least one pebble on $\Gamma(c, r+1)$ outside of $\Pi_{2r}^{(j)}$. To see why, note that if we can prove this, then simply by using the the fact that $\mathit{BW\text{-}Peb}^{\emptyset}(\Pi_{2r}) =$

$r+2$ and *BW-Peb*$^{\emptyset}(\Pi_{2r}) = 2r+2$ and adding one for the pebble outside of $\Pi_{2r}^{(j)}$ we have the matching lower bounds that we need. We present the argument for black-white pebbling, which is the harder case. The black-only pebbling case is handled completely analogously.

Suppose in order to get a contradiction that there is a visiting pebbling strategy $\mathcal{P}$ for $\Gamma(c, r+1)$ in space $r+2$. By Observation 2.2, $\mathcal{P}$ performs a complete visiting pebbling of every pyramid $\Pi_{2r}^{(j)}$. Consider the first time τ_1 when some pyramid has been completely pebbled and let this pyramid be $\Pi_{2r}^{(j_1)}$. Then at some time $\sigma_1 < \tau_1$ there are $r+2$ pebbles on $\Pi_{2r}^{(j_1)}$ and the rest of the graph $\Gamma(c, r+1)$ must be empty of pebbles at this point.

We claim that this implies that no vertex in $\Gamma(c, r+1)$ outside of the pyramid $\Pi_{2r}^{(j_1)}$ has been pebbled before time σ_1. Let us prove this crucial fact by a case analysis.

1. No vertex v in any other pyramid $\Pi_{2r}^{(j')}$ can have been pebbled before time σ_1. For if so, Lemma 9.8 says that $\Pi_{2r}^{(j')}$ has been completely pebbled before time σ_1, contradicting our choice of $\Pi_{2r}^{(j_1)}$ as the first such pyramid.

2. No vertex on a spine has been pebbled before time σ_1. This is so since Lemma 9.7 tells us that if some vertex on a spine has been pebbled, then the whole spine must have been pebbled in view of the fact that it is empty at time σ_1. But then Lemma 3.11 implies that all pyramid sinks must have been pebbled. This case has already been excluded.

3. Finally, no vertex v in $\Gamma(c, r)$ can have been pebbled before time σ_1. Otherwise the frugality of $\mathcal{P}$ implies (by pattern matching on the arguments in the proofs of Lemmas 3.11 and 9.7) that some successor of v must have been pebbled as well, and some successor of that successor et cetera, all the way up to where $\Gamma(c, r)$ connects with the spines. But we have ruled out the possibility that a spine vertex has been pebbled.

This establishes the claim, and we are now almost done. To clinch the argument, we need a couple of final observations. Note first that by frugality, at some time in the interval (σ_1, τ_1) some vertex in some spine must have been pebbled. This is so since the pyramid sink z_{j_1} has been pebbled before time τ_1, all of $\Pi_{2r}^{(j_1)}$ is empty at time τ_1, and all spines are empty at time $\sigma_1 < \tau_1$. But then Lemma 9.7 tells us that there will remain a pebble on this spine until all of the spine has been completely pebbled.

Consider now the second pyramid $\Pi_{2r}^{(j_2)}$ completely pebbled by $\mathcal{P}$, say, at time τ_2. At some point in time $\sigma_2 < \tau_2$ we have $r+2$ pebbles on $\Pi_{2r}^{(j_2)}$, and moreover $\sigma_2 > \tau_1$ since $\Pi_{2r}^{(j_2)}$ is empty at time τ_1. But now it must hold that either there is a pebble on a spine at this point, or, if all spines are completely empty, that some spine has been completely pebbled. If some spine has been completely pebbled, however, this in turn implies (appealing to Lemma 3.11 again) that there must be some pebble somewhere in some other pyramid $\Pi_{2r}^{(j')}$ at time σ_2. Thus the pebbling space exceeds $r+2$ and we have obtained our contradiction. The lemma follows. □

Studying the pebbling strategies in the proof of Lemma 9.3, it is not hard to see that they are very inefficient. The subgraphs in $\Gamma(c, r)$ will be pebbled over and over again, and for every step in the recursion the time required multiplies. We next show that if we are a bit more generous with the pebbling space, then we can get down to linear time.

Proof of Lemma 9.4. We want to prove that $\Gamma(c, r)$ has a persistent black pebbling strategy $\mathcal{P}$ that pebbles every vertex in $\Gamma(c, r)$ exactly once and uses space $\mathrm{O}(c+r)$. Suppose that there is such a pebbling strategy $\mathcal{P}_r$ for $\Gamma(c, r)$. We describe how to construct a pebbling $\mathcal{P}_{r+1}$ for $\Gamma(c, r+1)$ inductively. Note that the base case for $\Gamma(c, 1)$ is trivial.

The construction of $\mathcal{P}_{r+1}$ is very straightforward. First use $\mathcal{P}_r$ to make a persistent pebbling of $\Gamma(c, r)$ in space $\mathrm{O}(c + r)$. At the end of $\mathcal{P}_r$, we have c pebbles on the sinks $\gamma_1, \ldots, \gamma_c$. Keeping these pebbles in place, pebble the pyramids $\Pi_{2r}^{(1)}, \ldots, \Pi_{2r}^{(c)}$ persistently one by one in space $\mathrm{O}(r)$ with a strategy pebbling each vertex exactly once (for instance, by pebbling the pyramid bottom-up level by level). We leave pebbles on all pyramid sinks $z_1, \ldots, z_c$. This stage of the pebbling only requires space $\mathrm{O}(c + r)$ and at the end we have $2c$ black pebbles on all pyramid sinks $z_1, \ldots, z_c$ and all sinks $\gamma_1, \ldots, \gamma_c$ of $\Gamma(c, r)$. Keeping all these pebbles in place, we can pebble all c spines in parallel in linear time with $c + 1$ extra pebbles. Clearly, the same strategy works for $\Gamma^*(c, r)$ as well. □

To establish the trade-off in Theorem 9.5, Carlson and Savage prove a slightly stronger statement by induction over r.

Lemma 9.9 ([CS82]). *Suppose that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a conditional black pebbling on $\Gamma^*(c, r)$ such that:*

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

Then it holds that $\mathsf{time}(\mathcal{P}) = \tau - \sigma \geq \left(\frac{c-s}{s+1}\right)^r \cdot r!\,$.

Clearly, Theorem 9.5, follows from Lemma 9.9 by setting $\mathbb{P}_\sigma$ be the empty configuration without any pebbles.

Proof of Lemma 9.9. Let us write $T(c, r, s) = \left(\frac{c-s}{s+1}\right)^r \cdot r!\,$. We want to prove that any pebbling of $\Gamma^*(c, r)$ as stated in the lemma takes time $T(c, r, s)$. The base case for $\Gamma^*(c, 1)$ is immediate since at least $c - s$ pebble placements on sinks are needed.

Suppose that the lemma holds for $\Gamma^*(c, r - 1)$. At time σ, at least one of the $c > \mathsf{space}(\mathbb{P}_\sigma)$ pyramids $\Pi_{r-1}^{(j)}$ has to be empty. It takes r pebbles to block all paths from the sources to the sink in this pyramid, and thus $r + c - 1$ pebbles to block all paths from sources to sinks in all pyramids of $\Gamma^*(c, r)$. Since the available space is strictly less than $(r + 2) + s \leq r + c - 1$, at all times during the pebbling there is at least one source-to-sink path in some pyramid that is pebble-free. By the proof of Theorem 4.8, placing a black pebble on the sink of this pyramid is as expensive as pebbling the whole pyramid from scratch, i.e., it requires $r + 1$ pebbles.

More than $c - s$ spines are completely empty at time σ, and thus have to be completely pebbled. By Lemma 9.7, each spine is pebbled during a contiguous time interval. Consider the first section to be completely pebbled in any such spine. Let σ' be the time when a black pebble is placed on the successor $v[1]_c$ of the sink in the rightmost pyramid $\Pi_{r-1}^{(c)}$. At this point, all pyramid sinks have been pebbled, and at least one of them was pebbled while there was a pebble on the spine section and starting from a pebble configuration with some open source-to-sink path in the pyramid being open. By the discussion above, this requires $r + 1$ pebbles on the pyramids plus one pebble on the spine section, implying that at some time $\sigma_1 < \sigma'$ there were less than s pebbles on $\Gamma^*(c, r - 1)$. Let $\tau_1 > \sigma'$ be the time when the uppermost vertex $v[1]_{2c}$ in the section is pebbled. During $[\sigma', \tau_1] \subseteq [\sigma_1, \tau_1]$ all sinks of $\Gamma^*(c, r - 1)$ must be pebbled. We can now apply our induction hypothesis on $\Gamma^*(c, r - 1)$ and conclude that this takes time $T(c, r - 1, s)$.

Since we are considering black-only pebblings, it follows from (the proof of) Lemma 9.7 that there is at most one pebble on each spine at any one given time. Hence, we can repeat the above argument r times for the r sections of the spine and conclude that pebbling the whole spine requires time $r \cdot T(c, r - 1, s)$. Moreover, at most $s + 1$ spines can have pebbles on them simultaneously, since otherwise there are less than the $r+1$ pebbles left required to pebble the pyramid sinks when advancing along the spine sections. Let us focus on the strictly more than $c-s$ spines guaranteed to be completely empty at the start of the pebbling and order them with regard to the time of the first pebble placement on the spine. Consider the time when the $(s+2)$nd spine is pebbled. Then at least one of the $(s+1)$st spines must have been completely pebbled. We can repeat this argument for every group of $s+1$ spines for a total number of $\frac{c-s}{s+1}$ times. This show that the total time required for all of the spines is at least $\frac{c-s}{s+1} \cdot r \cdot T(c, r-1, s) = T(c, r, s)$ and the lemma follows by induction. $\square$

Proving the corresponding trade-off for black-white pebbling turns out to be more complex. We can no longer assume that the vertices in a section are pebbled sequentially, and it is no longer true that each spine can only contain at most one pebble at any given time in a frugal pebbling. Also, pebbles left on the right places in pyramids can make a black-white pyramid pebbling very much cheaper even if there are still open source-to-sink paths. These were all crucial components in the proof of Lemma 9.9. The overall proof structure is still the same in the black-white pebbling case, though, in that we use induction to prove a slightly stronger lemma than the theorem that we are after.

Lemma 9.10 ([Nor10a]). *Suppose that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling on $\Gamma(c, r)$ such that*

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

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

We will have to spend some time working on this lemma before the proof is complete. The plan of the proof is as follows. We first show that not too many pyramids can be pebbled simultaneously in a space-efficient pebbling. Next, we show that this is true for the spines as well. Given these two facts, we can prove that as a spine is pebbled, we have to alternate back and forth between time intervals when there are a lot of pebbles on some pyramid and time intervals when all sinks in $\Gamma(c, r)$ are pebbled. This will allow us to apply the induction hypothesis multiple times as in the proof of the trade-off for black pebbling.

Let us establish the two technical lemmas that upper-bound how many pyramids and spine sections can contain pebbles simultaneously at any one given time in a pebbling subjected to space constraints as in Lemma 9.10. The claims in the two lemmas are very similar in spirit, as are the proofs, so we state the lemmas together and then present the proofs together.

Lemma 9.11 ([Nor10a]). *Suppose that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling on $\Gamma(c, r)$ and that s is a constant satisfying the conditions in Lemma 9.10. Then at all times during the pebbling $\mathcal{P}$ strictly less than $4(s+1)$ pyramids $\Pi^{(j)}_{2r}$ contain pebbles simultaneously.*

Lemma 9.12 ([Nor10a]). *Suppose that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling on $\Gamma(c, r)$ and that s is a constant satisfying the conditions in Lemma 9.10. Then at all times during the pebbling $\mathcal{P}$ strictly less than $4(s+1)$ spine sections contain pebbles simultaneously.*

Note that Lemma 9.12 provides a total bound on the number of pebbled sections in all c spines. There might be spines containing several sections being pebbled simultaneously (in fact, this is exactly what one would expect a black-white pebbling to do to optimize the time given the space constraints), but what Lemma 9.12 says that if we fix an arbitrary time $t \in [\sigma, \tau]$, add up the number of sections containing pebbles at time t in each spine, and sum over all spines, we never exceed $4(s+1)$ sections in total at any point in time $t \in [\sigma, \tau]$.

Proof of Lemma 9.11. Suppose that on the contrary, there is some time $t^* \in (\sigma, \tau)$ when at least $4s+4$ pyramids $\Pi^{(j)}$ in $\Gamma(c,r)$ contain pebbles. Of these pyramids, at least $2s+4$ are empty both at time σ and at time τ since $\mathit{space}(\mathbb{P}_\sigma) < s$ and $\mathit{space}(\mathbb{P}_\tau) < s$. By Lemma 9.8, these pyramids, which we denote $\Pi^{(1)}, \ldots, \Pi^{(2s+4)}$, are completely pebbled during $[\sigma, \tau]$. Moreover, we can conclude that for every $\Pi^{(j)}$, $j = 1, \ldots, 2s+4$, there is an interval $[\sigma_j, \tau_j] \subseteq [\sigma, \tau]$ such that $t^* \in (\sigma_j, \tau_j)$ and $\Pi^{(j)}$ is empty at times σ_j and τ_j but contains pebbles throughout the interval (σ_j, τ_j) during which it is completely pebbled.

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

1. At least half of all t_j^* occur before (or at) time t^*, i.e., they satisfy $t_j^* \leq t^*$. If so, look at the largest $t_j^* \leq t^*$. At this time there are at least $r+1$ pebbles on $\Pi^{(j)}$ and at least $\frac{2s+4}{2} - 1 = s+1$ pebbles on other pyramids, which means that $\mathit{space}\left(\mathbb{P}_{t_j^*}\right) \geq (r+2)+s$. In other words, $\mathcal{P}$ exceeds the space restrictions in Lemma 9.10. Contradiction.

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

Hence, if $\mathcal{P}$ is a pebbling that complies with the restrictions in Lemma 9.10, it can never be the case that $4s+4$ pyramids $\Pi^{(j)}$ in $\Gamma(c,r)$ contain pebbles simultaneously. ☐

Proof of Lemma 9.12. Suppose that at some time $t^* \in (\sigma, \tau)$ at least $4s+4$ sections contain pebbles. At least $2s+4$ of these sections are empty at times σ and τ. Let us denote these sections $R_1, \ldots, R_{2s+4}$. Appealing to Lemma 9.7, we conclude that there are intervals $[\sigma_j, \tau_j] \subseteq [\sigma, \tau]$ for $j = 1, \ldots, 2s+4$, such that $t^* \in (\sigma_j, \tau_j)$ and R_j is empty at times σ_j and τ_j but contains pebbles throughout the interval (σ_j, τ_j) during which it is completely pebbled.

By Lemma 9.11, we know that less than $4s+4$ pyramids contain pebbles at time σ_j and similarly at time τ_j. Since all c pyramids in $\Gamma(c,r)$ must have their sinks pebbled during (σ_j, τ_j) but it holds that $2 \cdot (4s+4) < c$ by the assumptions in Lemma 9.10, we conclude from Lemma 9.8 that for every section R_j we can find some pyramid $\Pi^{(j)}$ that is completely pebbled during the interval (σ_j, τ_j). This, in turn, implies that there is some time $t_j^* \in (\sigma_j, \tau_j)$ when the pyramid $\Pi^{(j)}$ contains at least $\mathit{BW\text{-}Peb}^{\emptyset}\left(\Pi^{(j)}\right) = r+1$ pebbles. (We note that many t_j^* can be equal and even refer to the same pyramid, but this is not a problem.)

As in the proof of Lemma 9.11, we now sort the t_j^*, $j = 1, \ldots, 2s+4$, in increasing order and consider the two possible cases. If at least half of all t_j^* satisfy $t_j^* \leq t^*$, we look at the largest $t_j^* \leq t^*$. At this time there are at least $r+1$ pebbles on $\Pi^{(j)}$ and at least $\frac{2s+4}{2} = s+2$ pebbles on different sections, which means that $\mathit{space}\left(\mathbb{P}_{t_j^*}\right) \geq r+s+3$ exceeds the space restrictions. If, on the other hand, at least half of all t_j^* satisfy $t_j^* > t^*$, then for the smallest t_j^* larger than t^* we can again conclude that $\mathit{space}\left(\mathbb{P}_{t_j^*}\right) \geq r+s+3$, which is a contradiction. The lemma follows. ☐

Putting everything together, we are able to establish the black-white pebbling trade-off result.

Proof of Lemma 9.10. Suppose that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a conditional black-white pebbling on $\Gamma(c,r)$ pebbling all sinks and that $\max\left\{\mathit{space}(\mathbb{P}_\sigma), \mathit{space}(\mathbb{P}_\tau)\right\} < s$ and $\mathit{space}(\mathcal{P}) < (r+2)+s$ for $0 < s \leq c/8 - 1$. Let us define

$$T(c,r,s) = \left(\frac{c-2s}{4s+4}\right)^r \cdot r! \quad . \tag{9.2}$$

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

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

Assume that the lemma holds for $\Gamma(c, r-1)$ and consider any pebbling of $\Gamma(c, r)$. Less than $2s$ spines contain pebbles at time σ or time τ. All the other strictly more than $c-2s$ spines are empty at times σ and τ but must be completely pebbled during $[\sigma, \tau]$ by Lemma 3.11.

Consider the first time σ' when any spine gets a pebble for the first time. Let us denote this spine by Q'. By Lemma 9.7 we know that Q' contains pebbles during a contiguous time interval until it is completely pebbled and emptied at, say, time τ'. During this whole interval $[\sigma', \tau']$ less than $4s + 4$ sections contain pebbles at any one given time, so in particular less then $4s+4$ spines contain pebbles. Moreover, Lemma 9.7 says that every spine containing pebbles will remain pebbled until completed. What this means is that if we order the spines with respect to the time when they first receive a pebble in groups of size $4s + 4$, no spine in the second group can be pebbled until the at least one spine in the first group has been completed.

We remark that this divides the spines that are empty at the beginning and end of $\mathcal{P}$ into strictly more than $\frac{c-2s}{4s+4}$ groups. Furthermore, we claim that completely pebbling just one empty spine requires at least $r \cdot T(c, r-1, s)$ time steps. Given these two claims we are done, since by combining them we can deduce that the total pebbling time is lower-bounded by

$$\frac{c-2s}{4s+4} r \cdot T(c, r-1, s) = T(c, r, s) \tag{9.3}$$

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

It remains to establish the claim. To this end, fix any spine Q^* empty at times σ^* and τ^* but completely pebbled in $[\sigma^*, \tau^*]$. Consider the first time $\tau_1 \in [\sigma^*, \tau^*]$ when any section in Q^*, let us denote it by R_1, has been completely pebbled (i.e., , all vertices has been touched by pebbles but are now empty again). During $[\sigma^*, \tau_1]$ all pyramid sinks $z_1, \ldots, z_c$ are pebbled (Lemma 3.11), and since less than $2 \cdot (4s+4) < c$ pyramids contain pebbles at times σ^* or τ_1 (Lemma 9.11), at least one pyramid is pebbled completely (Lemma 9.8), which requires $r+1$ pebbles. Moreover, there is at least one pebble on R_1 during this whole interval. Hence, there is a time $\sigma_1 \in [\sigma^*, \tau_1]$ when there are strictly less than $(r+2) + s - (r+1) - 1 = s$ pebbles on $\Gamma(c, r-1)$. Also, at this time σ_1 less than $4s + 4$ sections contain pebbles (Lemma 9.12), and in particular this means that there are pebbles on less than $4s + 3$ other section of our spine Q^*. This puts an upper bound on the number of sections of Q^* pebbled this far, since every section is completely pebbled during a contiguous time interval before being emptied again, and we chose to focus on the first section R_1 in Q^* that was finished.

Look now at the first section R_2 in Q^* *other than the less than $4s + 4$ sections containing pebbles at time σ_1* that is completely pebbled, and let the time when R_2 is finished be denoted τ_2 (clearly, $\tau_2 > \tau_1$). During $[\sigma_1, \tau_2]$ all sinks of $\Gamma(c, r-1)$ must have been pebbled, and at time $\tau_2 - 1$ less than $4s + 3$ other section in Q^* contain pebbles.

Wrapping up, consider the first new section R_3 in our spine Q^* to be completely pebbled among those that has not yet been touched at time $\tau_2 - 1$. Suppose that R_3 is finished at time τ_3. Then during $[\tau_2, \tau_3]$ some pyramid is completely pebbled, and thus there must exist a time $\sigma_3 \in (\tau_2, \tau_3)$ when there are at least $r + 1$ pebbles on this pyramid and at least one pebble on the spine Q^*, leaving less than s pebbles for $\Gamma(c, r-1)$. But this means that we can apply the induction hypothesis on the interval $[\sigma_1, \sigma_3]$ and deduce that $\sigma_3 - \sigma_1 \geq T(c, r-1, s)$. Note also that at time σ_3 less than $8s + 8 < c$ sections in Q^* have been finished.

Continuing in this way, for every group of $8s + 8 < c$ finished sections in Q^* we get one pebbling of $\Gamma(c, r-1)$ in space less than $\textit{BW-Peb}^{\emptyset}\big(\Gamma(c, r-1)\big) + s$ and with less than s pebbles in the start and end configurations, which allows us to apply the induction hypothesis a total number of at least $\frac{cr}{8s+8} > r$ times.

(Just to argue that we get the constants right, note that $8s + 8 < c$ implies that after the final pebbling of the sinks of $\Gamma(c, r-1)$ has been done, there is still some empty section left in Q^*. When this final section is taken care of, we will again get at least $r+1$ pebbles on some pyramid while at least one pebble resides on Q^*, so we get the space on $\Gamma(c, r-1)$ down below s as is needed for the induction hypothesis.)

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

As we already noted, this completes the proof of Theorem 9.6, since this theorem follows immediately from Lemma 9.10 for the special case when $\mathbb{P}_\sigma = \mathbb{P}_\tau = (\emptyset, \emptyset)$.

10 Robust Time-Space Trade-offs

In the paper [CS80], Carlson and Savage initiated the study of *robust trade-offs*, by which we mean that the superpolynomial lower bound on pebbling should hold over as broad as possible a range of space. To get robust pebbling trade-offs, we can use the results in Section 9 extending [CS82], as well as a DAG family studied in [LT82, Section 4].

The first theorem below follows as a corollary of what we have already done in Section 9. We will spend the rest of this section describing the tools needed to establish the second theorem.

Theorem 10.1. *There are families of explicitly constructible families of single-sink DAGs $\{G_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

1. *Every graph G_n has black pebbling price $\mathit{Peb}(G_n) = \mathrm{O}(\log n)$.*
2. *There is a complete black pebbling of G_n in space $\mathrm{O}\left(\sqrt[3]{n/\log^2 n}\right)$ and time $\mathrm{O}(n)$.*
3. *There is a constant $K > 0$ such that any black-white pebbling $\mathcal{P}_n$ of G_n with space $\mathit{space}(\mathcal{P}_n) \leq K\sqrt[3]{n/\log^2 n}$ must have $\mathit{time}(\mathcal{P}_n) = n^{\Omega(\log\log n)}$.*

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

Proof. Consider the graphs $\Gamma(c, r)$ in Definition 9.2 with parameters chosen so that $c = 2^r$. Then the size of $\Gamma(c, r)$ is $\Theta\big(r^2 2^{3r}\big)$ by Lemma 9.3. Let $r(n) = \max\{r : r^2 2^{3r} \leq n\}$ and define the graph family $\{H_n\}_{n=1}^{\infty}$ by $H_n = \Gamma(2^r, r)$ for $r = r(n)$. Finally, construct the single-sink version $G_n = \widehat{H_n}$ of H_n.

Translating from G_n back to $\Gamma(c, r)$ we have parameters $r = \Theta(\log n)$ and $c = \Theta\big((n/\log^2 n)^{1/3}\big)$, so Lemma 9.3 yields that $\mathit{Peb}(G_n) = \mathrm{O}(\log n)$. Also, the persistent black pebbling of G_n in Lemma 9.4 has space $\mathrm{O}\big((n/\log^2 n)^{1/3}\big)$. Setting $s = c/8 - 1$ in Theorem 9.6 shows that there is a constant K such that if the space of a black-white pebbling $\mathcal{P}_n$ drops below $K \cdot (n/\log^2 n)^{1/3} \leq (r+2) + s$, then we must have

$$\mathit{time}(\mathcal{P}_n) \geq \mathrm{O}(1)^r \cdot r! = n^{\Omega(\log\log n)} \tag{10.1}$$

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

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

Theorem 10.2. *There are explicitly constructible families of single-sink DAGs $\{G_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

1. *Every graph G_n has black pebbling price $\mathit{Peb}(G_n) = \mathrm{O}(\log^2 n)$.*
2. *There is a complete black pebbling of G_n in space $\mathrm{O}(n/\log n)$ and time $\mathrm{O}(n)$.*

3. *There is a constant $K > 0$ such that any complete black-white pebbling $\mathcal{P}_n$ of G_n with $\mathsf{space}(\mathcal{P}_n) \leq Kn/\log n$ must have $\mathsf{time}(\mathcal{P}_n) = n^{\Omega(\log\log n)}$.*

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

We remark that the results in Theorem 10.2 are arguably slightly stronger than those in Theorem 10.1, but they require a very much more involved graph construction with worse hidden constants than the very simple and clean construction underlying Theorem 10.1.

10.1 Definition of Superconcentrator Stacks and Statement of Results

To obtain the trade-off result in Theorem 10.2, we study graphs built by stacking superconcentrators on top of one another as follows.

Definition 10.3 (Stack of superconcentrators ([LT82])). Let SC_m denote any (explicitly constructible) linear-size m-superconcentrator with bounded indegree and depth $\log m$. We let $\Phi(m, r)$ denote the graph constructed by placing r copies $SC^1_m, \ldots, SC^r_m$ of SC_m on top of one another, with the sinks $z^j_1, z^j_2, \ldots, z^j_m$ of SC^j_m connected to the sources $s^{j+1}_1, s^{j+1}_2, \ldots, s^{j+1}_m$ of SC^{j+1}_m by edges $\bigl(z^j_i, s^{j+1}_i\bigr)$ for all $i = 1, \ldots, m$ and all $j = 1, \ldots, r-1$.

Clearly, $\Phi(m, r)$ has size $\Theta(rm)$. Figure 23 gives a schematic illustration of the construction.

Theorem 10.4 ([LT82]). *Let $\Phi(m, r)$ denote a stack of (explicitly constructible) linear-size m-superconcentrator with bounded indegree and depth $\log m$. Then the following holds:*

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

Proof sketch. The upper bound on black pebbling price follows from Observation 3.3, since the depth of $\Phi(m, r)$ is $\mathrm{O}(r \log m)$.

The linear-time black pebbling strategy is obtained by applying the trivial pebbling strategy in Observation 3.1 consecutively to each superconcentrator, keeping pebbles on the sinks of SC^j_m while pebbling SC^{j+1}_m.

The reason that the final trade-off result holds is, very loosely put, that the lower bounds in Lemma 5.4 and Theorem 5.5 propagate through the stack of superconcentrators and get multiplied at each level. If the pebbling strategy is restricted to keeping s/r pebbles on each copy SC^j_m of the superconcentrator, this is not hard to prove directly from Lemma 5.4. Establishing that this intuition holds also in the general case, when pebbles may be unevenly distributed over the superconcentrator copies, is much more technically challenging, however. $\square$

Now Theorem 10.2 follows by studying (single-sink versions of) superconcentrator stacks as in Definition 10.3 with $r = \Theta(\log n)$ and $m = \Theta(n/\log n)$ and applying Theorem 10.4.

10.2 Proofs of Superconcentrator Stack Properties

Edit comment 1: *. . . And here the intention is to provide if not all the details, then at least a fairly detailed sketch of the proof.*

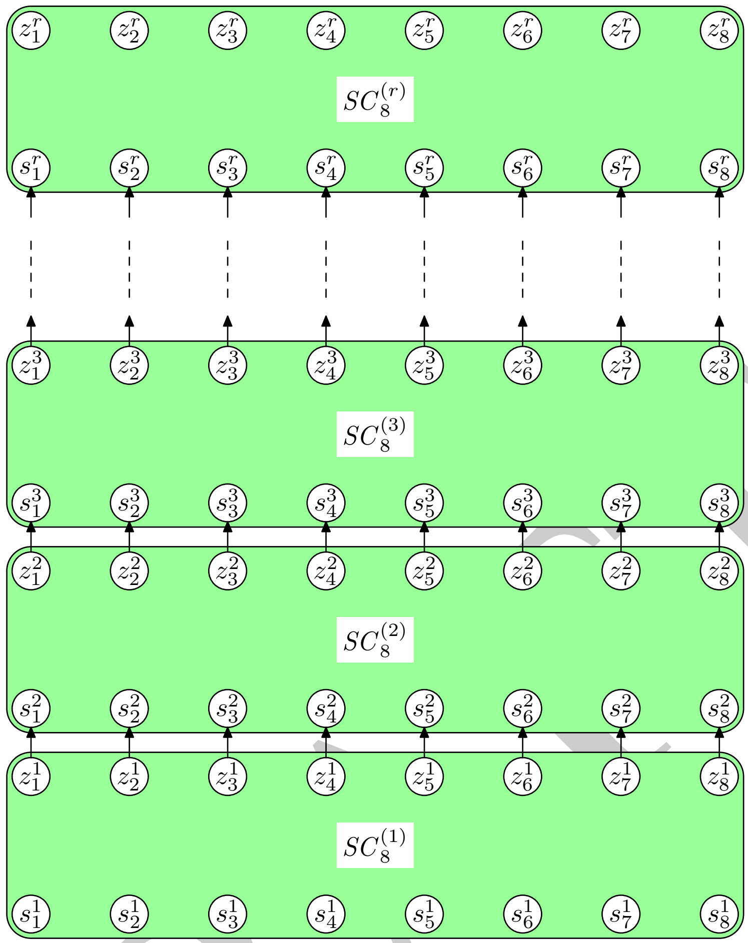


Figure 23: Schematic illustration of stack of superconcentrators $\Phi(8, r)$.

11 Exponential Time-Space Trade-offs

To get exponential trade-offs, i.e., trade-offs with lower bounds on the length on the form $\exp(n^{\epsilon})$ for some constant $\epsilon > 0$, the graphs in Section 10 are not sufficient. This again follows from the counting argument in the introduction, since no graph that can be pebbled in polylogarithmic space can require more than quasi-polynomial time for such a pebbling. Instead, we will consider graphs with pebbling price growing like n^k for some constant $k > 0$.

A first, easy exponential trade-off, which also exhibits a certain robustness, can be derived from the Carlson Savage DAGs Definition 9.2 studied in Section 9.

Theorem 11.1. *There are explicitly constructible families of single-sink DAGs $\{G_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

1. *Peb*$(G_n) = \mathrm{O}\left(\sqrt[8]{n}\right)$.
2. *There is a linear-time complete black pebbling of G_n in space $\mathrm{O}\left(\sqrt[4]{n}\right)$.*
3. *There is a constant $K > 0$ such that any complete black-white pebbling $\mathcal{P}_n$ of G_n in space at most $K\sqrt[4]{n}$ must take time $\left(\sqrt[8]{n}\right)!$.*

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

Proof. In the same way as in the proofs of Theorems 9.1 and 10.1, appeal to Theorem 9.6, this time with the parameter settings $c = \sqrt[4]{n}$ and $r = \sqrt[8]{n}$, and consider the single-sink versions of this graph family. $\square$

We remark that there is nothing magic in our particular choice of parameters c and r in Theorem 11.1. Other parameters could be plugged in instead and yield slightly different results.

Now that we know that there are robust exponential trade-offs, we want to obtain exponential trade-offs for the pebbling space being as large as possible. The trade-off in Theorem 11.1 cannot be pushed much higher than space $\Theta\big(\sqrt[4]{n}\big)$, but the upper bound in Theorem 6.2 a priori allows for exponential trade-offs all the way up to $\mathrm{O}(n/\log n)$.

The second, more challenging result in this section yields exponential trade-offs for space as high as $\Theta(n/\log n)$. This result is from [LT82, Section 5]. By stacking superconcentrators of *defferent sizes* on top of one another, Lengauer and Tarjan are able to prove a lower bound matching the upper bound in Theorem 6.2.

EDIT COMMENT 2: *Double-check the statement of Theorem 11.2 below.*

Theorem 11.2 ([LT82]). *There exist constants $\epsilon, K > 0$ such that for all sufficiently large integers n, s satisfying $Kn/\log n \leq s \leq n$, we can find an explicitly constructible single-sink DAG G with indegree 2 and number of vertices at most n such that any black-white pebbling strategy $\mathcal{P}$ for G with $\mathsf{space}(\mathcal{P}) \leq s$ must have $\mathsf{time}(\mathcal{P}) \geq s \cdot 2^{2^{\epsilon n/s}}$.*

Note that the graph G in Theorem 11.2 depends on the pebbling space parameter s. Lengauer and Tarjan conjecture that no single graph gives an exponential time-space tradeoff for the whole range of $s \in [n/\log n, n]$, but to the best of our knowledge this problem is still open.

Open Problem 7. *Are there graphs with time-space trade-offs following the whole (or at least parts of) the curves specified by Theorems 6.2 and 11.2, or can these trade-off curves only be realized as the collection of threshold trade-offs for many different individual graphs?*

EDIT COMMENT 3: *The intention here is to give a reasonably detailed exposition of [LT82, Section 5], at least describing the construction in full detail and then at least sketching the main ideas in the proofs (depending on how gory the analysis gets). All this remains to be done, however.*

12 Separations of Black and Black-White Pebbling

For almost all graph families presented in this survey, it is known that the black and black-white pebbling prices coincide asymptotically. It is natural to ask whether this is always the case or whether the nondeterminism of the white pebbles adds substantial extra computing power.

Meyer auf der Heide [Mey81] established an upper bound on what can be gained from white pebbles by proving that the difference in black and black-white pebbling price can be at most quadratic.

Theorem 12.1 ([Mey81]). *For any DAG G with black-white pebbling price $\mathsf{BW\text{-}Peb}^{\bullet}(G) \leq s$ it holds that the black pebbling price is at most $\mathsf{Peb}^{\bullet}(G) \leq (s^2 - s)/2 + 1$.*

For quite some time, there were not even any DAGs known for which the black and black-white pebbling prices differed by more than a constant factor. The first progress towards proving a lower bound matching the upper bound in Theorem 12.1 was made by Wilber [Wil88], who obtained the following asymptotical separation between black and black-white pebbling with respect to space.

Theorem 12.2 ([Wil88]). *There is a family $\{G(s)\}_{s=1}^{\infty}$ of DAGs of indegree 2 and size polynomial in s such that $\mathsf{BW\text{-}Peb}^{\bullet}(G(s)) = \mathrm{O}(s)$ but $\mathsf{Peb}^{\bullet}(G(s)) = \Omega(s \log s/\log\log s)$.*

This result was later improved by Kalyanasundaram and Schnitger [KS91] to a quadratic separation, which by Theorem 12.1 is optimal up to constant factors.

Theorem 12.3 ([KS91]). *There is a family $\{G(s)\}_{s=1}^{\infty}$ of DAGs of indegree 3 and size $\exp(\Theta(s \log s))$ such that $\textsf{BW-Peb}^{\bullet}(G(s)) \leq 3s + 1$ but $\textsf{Peb}^{\bullet}(G(s)) \geq s^2$.*

Note, hower, that the graphs yielding the optimal quadratic separation are not of size polynomial in s, as opposed to Wilber's result that holds for polynomial-size graphs. It would be interesting to know whether a quadratic separation can be obtained for graphs of polynomial size.

Open Problem 8 ([KS91]). *Is it possible to prove a quadratic separation between black and black-white pebbling as in Theorem 12.3 but for DAGs of size polynomial in the pebbling price?*

Kalyanasundaram and Schnitger also gave a theorem quantifying how many black pebbles are required to compensate for the loss of white pebbles.

Theorem 12.4 ([KS91]). *Any black-white pebbling strategy that uses at most q white pebbles and p black pebbles can be simulated by a black-only pebbling strategy that uses at most $(p + 3q/2)(q + 1)$ pebbles.*

We remark that both Theorem 12.1 and Theorem 12.4 have an exponential blowup in the pebbling time when going from black-white to black-only pebbling, so these theorems are of no help when we want to prove time-space trade-offs with upper bounds for black pebbling and lower bounds for black-white pebbling (as will be of interest in Section 14).

Below, we show Theorems 12.2 and 12.3, while referring the reader interested in the proofs of Theorems 12.1 and 12.4 to the original papers. We first do the proof of Theorem 12.3 in Section 12.1, and then in Section 12.2 give an exposition of the result in Theorem 12.2 as proven by [KS91], avoiding the ingenious but complicated construction in the original paper [Wil88].

12.1 A Quadratic Separation Between Black and Black-White Pebbling Space

The purpose of this section is to prove the theorem below, which implies Theorem 12.3 as a special case.

Theorem 12.5 ([KS91]). *There is a family $K(p, q)$ of DAGs of indegree 3 and size $\Theta\big((p+1)^{q+1}\big)$ such that $K(p, q)$ can be pebbled by a black-white pebbling using q white pebbles and $p + q + 1$ black pebbles, but any black-only pebbling strategy requires at least pq pebbles.*

In particular, setting $p = q$ it holds that $\textsf{Peb}^{\bullet}(K(p, q)) = \Omega\big(\textsf{BW-Peb}^{\bullet}(K(p, q))^2\big)$.

The definition of the graph family $K(p, q)$ is by induction. The base case graph is simply a line.

Definition 12.6 (m-line and $K(p, 0)$-graph [KS91]). An *m-line* is a DAG with vertex set $v_1, v_2, \ldots, v_m$ and edge set $\{(v_i, v_{i+1}) \mid i = 1, 2, \ldots, m-1\}$. The *ith column* of the m-line is the vertex v_i.

The graph $K(p, 0)$ is a p-line. We say that the *first row* $f_1, f_2, \ldots, f_p$ of $K(p, 0)$ and the *last row* $l_1, l_2, \ldots, l_p$ are both equal to $v_1, v_2, \ldots, v_p$.

In the general case of the definition, the graph $K(p, q)$ consists of a number of identical blocks $M(p, q)$, where each block contains a copy of $K(p, q-1)$. In the recursive constructions below, we will be somewhat sloppy with the indices in order not to clutter the notation. In particular, if we wanted to be formally correct, all subgraphs and vertices below should be labelled by their "level of recursion" q within the construction, as well as by a number indicating which of the identical copies on recursion level q the vertex resides in, but we believe that adding these extra indices would lead to more confusion than clarity.

The next definition is illustrated in Figure 24. We remark that the graph construction has been slightly modified as compared to [KS91].

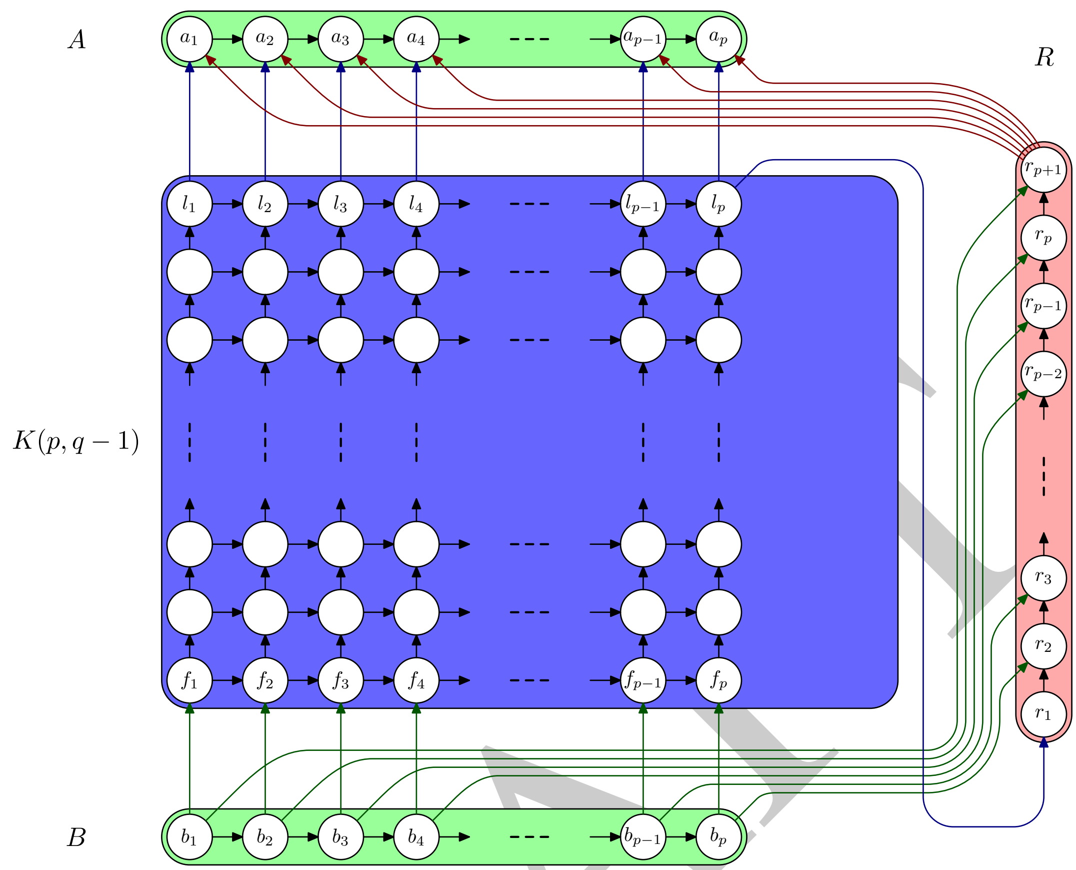


Figure 24: Building block $M(p, q)$ in graph $K(p, q)$ yielding quadratic pebbling price separation.

Definition 12.7 (**$M(p, q)$-block [KS91]**). Suppose that $K(p, q-1)$ has been defined. The *block graph* $M(p, q)$ consists of the following components:

- a copy of $K(p, q-1)$ with first row $f_1, f_2, \ldots, f_m$ and last row $l_1, l_2, \ldots, l_m$,
- a p-line B on vertices $b_1, b_2, \ldots, b_p$ before $K(p, q-1)$,
- a p-line A on vertices $a_1, a_2, \ldots, a_p$ after $K(p, q-1)$, and
- a $(p+1)$-line R on vertices $r_1, r_2, \ldots, r_{p+1}$ to the right of $K(p, q-1)$.

The subgraph components are connected by edges as follows:

- (b_i, f_i) for $i = 1, 2, \ldots, p$,
- (b_i, r_{p+2-i}) for $i = 1, 2, \ldots, p$,
- (l_i, a_i) for $i = 1, 2, \ldots, p$,
- a single edge (l_p, r_1), and
- (r_{p+1}, a_i) for $i = 1, 2, \ldots, p$.

The ith column of $M(p, q)$ consists of the ith column of $K(p, q-1)$ together with the vertices b_i and a_i.

Note that the vertices in R are not part of any column. Also note that Figure 24 only displays the vertices of the p-lines in $K(p, q-1)$ but omits all the copies of R which are there by recursion (they are supposed to reside in the shaded area on the right of the drawn vertices).

Definition 12.8 ($K(p, q)$-graph [KS91]). The graph $K(p, q)$ consists of $p+1$ copies of the block graph $M(p, q)$, which we denote $M^{(1)}(p, q), M^{(2)}(p, q), \ldots, M^{(p+1)}(p, q)$. The edges between the blocks are $\left(a_i^{(j)}, b_i^{(j+1)}\right)$ for $i, j = 1, 2, \ldots, p$, i.e., the last vertex in every column i in $M^{(j)}(p, q)$ is connected to the first vertex in the same column in $M^{(j+1)}(p, q)$.

We define the first row $f_1, f_2, \ldots, f_m$ of $K(p, q)$ to consist of the vertices $b_1^{(1)}, b_2^{(1)}, \ldots, b_p^{(1)}$ of the first block $M^{(1)}(p, q)$, and the last row $l_1, l_2, \ldots, l_m$, to consist of the vertices $a_1^{(p+1)}, a_2^{(p+1)}, \ldots, a_p^{(p+1)}$ of the last M-block. The ith column of $K(p, q)$ is the union of the ith columns of all the M-blocks.

Definition 12.8 is illustrated in Figure 25. It is clear that the indegree of $K(p, q)$ is 3 and it is also straightforward to verify the size bound in Theorem 12.5. We therefore focus on the upper and lower bounds on pebbling price. We attend to the upper bound first.

Lemma 12.9 ([KS91]). *The graph $K(p, q)$ has a persistent black-white pebbling strategy using q white pebbles and $p+q+1$ black pebbles.*

Proof. The proof is by induction over q. For the base case, note that we can place black pebbles on all vertices in $K(p, 0)$ in space p, since $K(p, 0)$ is just a p-line.

Inductively, suppose that we have constructed for $K(p, q-1)$ a black-white pebbling $\mathcal{P}$ starting with $(B_\sigma, W_\sigma) = (\{f_1, f_2, \ldots, f_p\}, \emptyset)$, i.e., black pebbles on all vertices in the first row and no white pebbles, and ending with $(B_\tau, W_\tau) = (\{l_1, l_2, \ldots, l_p\}, \emptyset)$, i.e., black pebbles on all vertices in the last row and no white pebbles. Suppose furthermore that $\mathcal{P}$ never uses more than $q-1$ white pebbles and $p+q$ black pebbles.

We want to show how to construct a pebbling $\mathcal{P}'$ for the block graph $M(p, q)$ starting with $(B_{\sigma'}, W_{\sigma'}) = (\{b_1, b_2, \ldots, b_p\}, \emptyset)$, ending with $(B_{\tau'}, W_{\tau'}) = (\{a_1, a_2, \ldots, a_p\}, \emptyset)$, and using no more than q white pebbles and $p+q+1$ black pebbles. It is easy to see that given such a pebbling $\mathcal{P}'$ for $M(p, q)$, we can extend it to a pebbling from a black-pebbled first row in $K(p, q)$ to a black-pebbled last row simply by pebbling the blocks $M^{(j)}(p, q)$, $j = 1, 2, \ldots, p+1$ one by one in $p+1$ phases using $\mathcal{P}'$ repeatedly, and shifting the black pebbles from $a_i^{(j)}$ to $b_i^{(j+1)}$, $i = 1, 2, \ldots, p$, between each phase.

Thus, suppose that we have the pebble configuration $(B_{\sigma'}, W_{\sigma'}) = (\{b_1, b_2, \ldots, b_p\}, \emptyset)$ in $M(p, q)$. Place a white pebble on r_1. Thanks to this pebble and the black one on b_p, we can black-pebble r_2, and then black-pebble r_3 with the help of the pebbles on r_2 and b_{p-1}. Continue moving the two black pebbles on R all the way up until we have a single black pebble on r_{p+1}, which is possible in view of the pebbles on $b_{p-2}, \ldots, b_1$. This requires a total of $p+3$ pebbles.

Next, shift the black pebbles from b_i to f_i for all $i \in [p]$ in increasing order of the index i. This requires one additional auxiliary black pebble, and at the end of this step we have pebbles on $f_1, f_2, \ldots, f_p$ plus one black and one white pebble on R.

Now appeal to the inductive hypothesis to obtain a pebbling moving the black pebbles in $K(p, q-1)$ from $f_1, f_2, \ldots, f_p$ to $l_1, l_2, \ldots, l_p$. Adding the two pebbles on R, we get that this part of the pebbling uses at most q white pebbles and $p+q+1$ black pebbles.

To conclude the pebbling of $M(p, q)$, remove the white pebble on r_1, which is possible since l_p now has a pebble. Finally, shift the black pebbles from l_i to a_i for all $i \in [p]$ with the help of the black pebble on r_{p+1} and one auxiliary pebble, and then remove the pebble on r_{p+1}. This completes the induction step, and the lemma follows. □

To prove the lower bound on black-only pebbling space in Theorem 12.5, the following notation will be convenient.

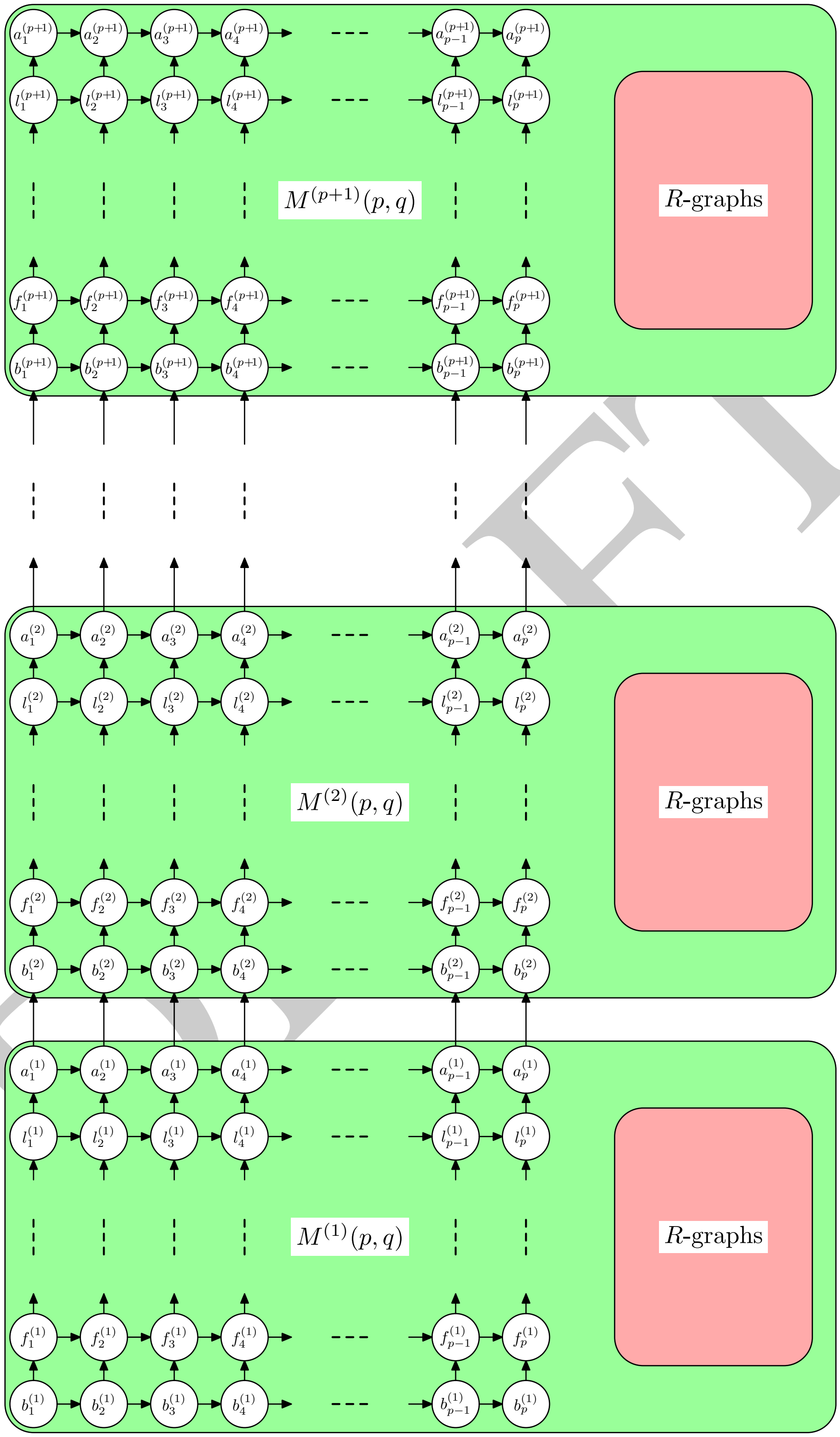


Figure 25: Building blocks $M(p,q)$ connected to form graph $K(p,q)$.

Definition 12.10 (Subgraphs $K(p,q)[m]$ and $K^{+B}(p,q)[m]$ [KS91]). We write $K(p,q)[m]$ to denote the subgraph of $K(p,q)$ induced on the m first blocks of the graph, i.e., the subgraph consisting of the blocks $M^{(1)}(p,q), M^{(2)}(p,q), \ldots, M^{(m)}(p,q)$ with edges $\left(a_i^{(j)}, b_i^{(j+1)}\right)$ for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, m-1$ between the blocks.

We let $K^{+B}(p,q)[m]$ denote the subgraph consisting of the m first blocks plus the first p-line B of the $(m+1)$st block of $K(p,q)$, i.e., the subgraph induced on $M^{(1)}(p,q) \cup \ldots \cup M^{(m)}(p,q) \cup B^{(m+1)}$.

Lemma 12.11 ([KS91]). *Suppose for $q \geq 1$ and $1 \leq m \leq p+1$ that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a black pebbling on $K(p,q)[m]$ such that some column c is completely free of pebbles at time σ but the last vertex $a_c^{(m)}$ in column c is pebbled at time τ. Then $\mathsf{space}(\mathcal{P}) \geq p(q-1)+m$.*

In particular, any complete black pebbling strategy for $K(p,q)$ requires space at least pq.

In should be noted that no assumptions are made about $\mathbb{P}_\sigma$ other than that column c is pebble-free. Thus, there may be pebbles placed strategically on other vertices at time σ, and the lower bound proof has to work even in the presence of such pebbles.

Before presenting a rigorous proof of Lemma 12.11, let us try to provide some intuition why it should be true. Note that if there were no R-graphs in $K(p,q)$ but only the p columns, then it would be straightforward to do a complete bottom-up pebbling with just $p+1$ black pebbles. Indeed, one way of viewing the pebbling strategy for the blocks $M(p,q)$ in the proof of Lemma 12.9 is that by keeping $2q$ pebbles on the relevant R-graphs, it becomes possible to implement this simple bottom-up strategy using only black pebbles on the columns of $K(p,q)$.

The crucial observation is that the bottom-up approach for $M(p,q)$ no longer works if we only have access to black pebbles. In this case, to pebble R we cannot rely on white pebbles but instead have to black-pebble all the way to the rightmost vertex l_p in the last row of $K(p,q-1)$ in order to reach r_1. But then to advance the pebble along $r_2, r_3, \ldots$ up to r_{p+1}, we need to have saved the pebbles on $b_1, b_2, \ldots, b_p$, or to repebble these vertices while keeping the pebbles on $l_1, l_2, \ldots, l_p$. In both cases we incur an extra space penalty as compared to the black-white pebbling strategy, and this penalty is compounded at each level of the recursion. This is, loosely speaking, what leads to the blow-up in pebbling space. The formal details of the argument follow.

Proof of Lemma 12.11. The proof is by induction over q and m. The base case $q = m = 1$ is immediate, since at least one pebble is clearly needed.

First induction step: Suppose that the lemma holds for $1 \leq q' < q$ and also for $q' = q$ and $1 \leq m' \leq m < p+1$. We want to show that it is true for q and $m+1$, i.e., for the graph $K(p,q)[m+1]$.

Let $I_1 = [\sigma_1, \tau_1] \subseteq [\sigma, \tau]$ be the time interval such that the very first vertex in the cth column $b_c^{(1)}$ is pebbled for the first time at time $\sigma_1 + 1$ and the last vertex $a_c^{(m)}$ in the cth column of $K(p,q)[m]$ is pebbled for the first time at time τ_1. Such a time interval clearly exists by Observation 3.10, and by the same observation the column c is pebble-free at time σ_1, so the induction hypothesis applies to $K(p,q)[m]$ and the subpebbling $\mathcal{P}_1 = \{\mathbb{P}_{\sigma_1}, \ldots, \mathbb{P}_{\tau_1}\}$. We get the following case analysis.

Case 1: There is always at least one pebble in $M^{(m+1)}(p,q)$ throughout the time interval I_1. If so, the induction step follows and we are done.

Case 2: There is a time t_1 in $[\sigma_1, \tau_1]$ when $M^{(m+1)}(p,q)$ is completely pebble-free. Observe that this implies that all columns numbered c and higher in $M^{(m+1)}(p,q)$ must be empty at time τ_1, since they cannot be pebbled after t_1 before a pebble is placed on $a_c^{(m)}$, the last vertex in column c in $M^{(m)}(p,q)$. Let t' be the last time before τ when $M^{(m+1)}(p,q)$ is completely empty of pebbles, and let $t'' = \max\{t', \tau_1\}$. Consider the time interval $I_2 = [\sigma_2, \tau_2]$ such that $b_c^{(m+1)}$ is pebbled for the first time after t'' at time $\sigma_2 + 1$ and $l_c^{(m+1)}$ is pebbled for the first time after σ_2 at time τ_2. Note that by construction, $M^{(m+1)}(p,q)$ contains at least one pebble from time $\sigma_2 + 1$ all the way up to time τ when the final black pebble is placed on $a_c^{(m+1)}$.

Recall that $K^{+B}(p,q)[m]$ is the subgraph induced on $M^{(1)}(p,q) \cup \ldots \cup M^{(m)}(p,q) \cup B^{(m+1)}$. We split the further analysis into two subcases.

Case 2a: Suppose that during all of I_2 there is at least one pebble per column on $K^{+B}(p,q)[m]$. The pebbling of the cth column in the subgraph $K^{(m+1)}(p,q-1)$ of the $(m+1)$st block, which certainly has to happen during I_2 in order to get a pebble on $l_c^{(m+1)}$, requires $p(q-2)+p+1$ pebbles if $q > 1$ by the induction hypothesis, and if $q = 1$ it is clear that the cth column of $K(p,0)$ requires at least 1 pebble. Hence, adding up all pebbles we get a total of $p(q-1)+p+1 \geq p(q-1)+m+1$, and the induction step follows.

Case 2b: If case 2a does not apply, then there exists a time $t_2 \in [\sigma_2, \tau_2]$ when some column d of $K^{+B}(p,q)[m]$ is pebble-free. We claim that this column must be completely pebbled during $[t_2, \tau]$.

Taking this claim on faith for the moment, we observe that this is sufficient to prove the induction step. For, in particular, this means that the dth column of $K(p,q)[m]$ is pebbled during $[t_2, \tau]$, which by the induction hypothesis requires $p(q-1)+m$ pebbles. Since in addition $\sigma_2 < t_2$ was chosen so that the $(m+1)$ block $K(p,q)[m+1]$ contains at least one pebble at all times after σ_2, we get a total of at least $p(q-1)+m+1$ pebbles as required.

It remains to prove the claim. Note that so far, we have not used the R-graphs at all, but it is clear from the discussion preceding the proof of of the lemma that these graphs have to play a crucial role in any lower bound proof. Focusing on the the subgraph $R^{(m+1)}$, therefore, we first recall that all of $M^{(m+1)}(p,q)$, including $R^{(m+1)}$, is empty of pebbles at time t' and that as a consequence of this all columns $c' \geq c$ in $K(p,q)[m+1]$ must be pebble-free throughout $[t', \sigma_2]$. Furthermore, all vertices $l_c^{(m+1)}, l_{c+1}^{(m+1)}, \ldots l_p^{(m+1)}$ must remain empty until at least time $\tau_2 - 1$. This implies that $R^{(m+1)}$ is still completely free of pebbles at time τ_2. But $R^{(m+1)}$ must be pebbled before a pebble can be placed on $a_c^{(m+1)}$, since there is an edge from $r_{p+1}^{(m+1)}$ to $a_c^{(m+1)}$. This means that all predecessors of $R^{(m+1)}$ in $B^{(m+1)}$ have to contain pebbles as well at some point after τ_2, which implies that a pebble is placed on $b_d^{(m+1)}$ after time τ_2. But this vertex is the last one in column d of $K^{+B}(p,q)[m]$. It follows that the whole column d of $K^{+B}(p,q)[m]$ is indeed pebbled during $[t_2, \tau]$ as claimed.

Second induction step: Assume now that the lemma holds for all $q' \leq q$ and all $m' \leq p+1$. We want to prove it for $q' = q+1$ and $m' = 1$. But this step is immediate, since $K(p,q+1)[1]$ contains a copy of $K(p,q)$, and by the induction hypothesis pebbling a column in this graph requires at least $p(q-1)+p+1 = pq+1$ pebbles. The lemma follows. □

This concludes the proof of Theorem 12.5, which also establishes the quadratic separation between black-white and black-only pebbling in Theorem 12.3 as a special case.

12.2 A Weaker Separation for Polynomial-Size Graphs

The main drawback of Theorem 12.3, as pointed out by the authors themselves in [KS91], is that the graphs needed to obtain a quadratic separation of black and black-white pebbling are of size exponential in the pebbling price. Somehow, it would be preferable to display this kind of separation for graphs where the pebbling price and graph size parameters are polynomially related.

If we insist on proving separations for graphs where the size is polynomial in the pebbling price, the best known separation is still the result in [Wil88]. We present a proof of this separation below, but instead of the original (clever but quite involved) construction in [Wil88] we follow the alternative proof in [KS91] of a result with very similar parameters, which generalizes the construction in Section 12.1.

Theorem 12.12 ([KS91]). *There is a family of directed acyclic graphs $\Lambda(p,q,k)$, for $q \leq p$ and $k \leq p$, of indegree 3 and size $\mathrm{O}\big(\mathrm{poly}(p)(p/k)^q\big)$ such that $\Lambda(p,q,k)$ can be pebbled by a persistent black-white pebbling using kq white pebbles and $p+kq+1$ black pebbles, but any black-only pebbling strategy requires at least pq pebbles.*

In particular, setting $k = p \log\log p / \log p$ and $q = \log p / \log\log p$, it holds that the graphs have size $\mathrm{poly}(p)$*, black pebbling price* $\mathit{Peb}^{\bullet}(\Lambda(p,q,k)) = \Omega(p \log p / \log\log p)$*, and black-white pebbling price* $\mathit{BW\text{-}Peb}^{\bullet}(\Lambda(p,q,k)) = \mathrm{O}(p)$.

To present the construction of $\Lambda(p,q,k)$, we first need to generalize Definition 12.6 as follows.

Definition 12.13 (Spiral mesh and $\Lambda(p,0,k)$-graph). An (n,m)-*spiral mesh* is a directed acyclic graph on vertices $\{v_{i,j} \mid i \in [n], j \in [m]\}$ with edges $(v_{i,j}, v_{i,j+1})$ for $i \in [n]$ and $j \in [m-1]$, $(v_{i,j}, v_{i+1,j})$ for $i \in [n-1]$ and $j \in [m]$, and $(v_{i,m}, v_{i+1,1})$ for $i \in [n-1]$. The *ith column* of the (n,m)-spiral mesh consists of the vertices $v_{i,j}$ for $j \in [m]$.

The graph $\Lambda(p,0,k)$ is a $(1,p)$-mesh, i.e., a p-line, the *first row* $f_1, f_2, \ldots, f_p$ and *last row* $l_1, l_2, \ldots, l_p$ of which are defined to be the vertices in the graph.

The definition of this graph family, too, is by induction, where $\Lambda(p,q,k)$ consists of a number of identical building blocks which all contain a copy each of $\Lambda(p,q-1,k)$. We give the definition of these building block graphs next, continuing our policy of being somewhat sloppy with indices to avoid cluttering the notation. A pictorial representation of the definition is given in Figure 26, where again the graph construction has been slightly modified as compared to [KS91].

Definition 12.14 ($N(p,q,k)$-block [KS91]). Suppose that $\Lambda(p,q-1,k)$ has been defined. The *block graph* $N(p,q,k)$, where $k \le p$, consists of the following components:

- a copy of $\Lambda(p,q-1,k)$ with first row $f_1, f_2, \ldots, f_m$ and last row $l_1, l_2, \ldots, l_m$,
- a $\bigl((p+1)^2, p\bigr)$-spiral mesh B on vertices $b_{i,j}$, $i \in \bigl[(p+1)^2\bigr]$, $j \in [p]$,
- a $\bigl((p+1)^3, p\bigr)$-spiral mesh A on vertices $a_{i,j}$, $i \in \bigl[(p+1)^3\bigr]$, $j \in [p]$,
- k copies $R_1, \ldots, R_k$ of a $(p+1)$-line, with the ith copy having vertices $r_{i,j}$ for $j \in [p+1]$.

For ease of notation, in what follows we will write $n_b = (p+1)^2$ and $n_a = (p+1)^3$ for the number of rows in B and A.

The subgraph components are connected by edges as follows (where we use the notation $\bigl(u;v\bigr)$ for the edge from u to v for clarity):

- $\bigl(b_{n_b,j}\,;\,f_j\bigr)$ for $j \in [p]$,
- $\bigl(b_{n_b,j}\,;\,r_{i,p+2-j}\bigr)$ for $i \in [k]$ and $j \in [p]$,
- $\bigl(l_j; a_{1,j}\bigr)$ for $j \in [p]$,
- $\bigl(l_{\lfloor ip/k \rfloor}; r_{i,1}\bigr)$ for $i \in [k]$, and
- $\bigl(r_{i,p+1}\,;\,a_{1,j}\bigr)$ for all $i \in [k]$ and all j such that $(i-1)p/k < j \le ip/k$.

The ith column of $N(p,q,k)$ consists of the ith columns of B, $\Lambda(p,q-1,k)$, and A.

Definition 12.15 ($\Lambda(p,q,k)$-graph [KS91]). The graph $\Lambda(p,q,k)$ consists of $\lceil p/k \rceil + 1$ copies of the block graph $N(p,q,k)$, which we denote $N^{(1)}(p,q,k), N^{(2)}(p,q,k), \ldots, N^{(\lceil p/k \rceil + 1)}(p,q,k)$. The edges between the blocks are $\bigl(a^{(i)}_{n_a,j}\,;\,b^{(i+1)}_{1,j}\bigr)$ for $i = 1, \ldots, \lceil p/k \rceil$ and $j = 1, \ldots, p$, i.e., the last vertex in every column in the ith N-block is connected to the first vertex in the same column in the $(i+1)$st N-block.

We define the first row $f_1, f_2, \ldots, f_m$ of $\Lambda(p,q,k)$ to consist of the first row $b^{(1)}_{1,1}, b^{(1)}_{1,2}, \ldots, b^{(1)}_{1,p}$, of the first N-block and the last row $l_1, l_2, \ldots, l_m$, to consist of the last row $a^{(\lceil p/k \rceil+1)}_{n_a,1}, a^{(\lceil p/k \rceil+1)}_{n_a,2}, \ldots, a^{(\lceil p/k \rceil+1)}_{n_a,p}$ of the last N-block. The ith column of $\Lambda(p,q,k)$ is the union of the ith columns of all the N-blocks.

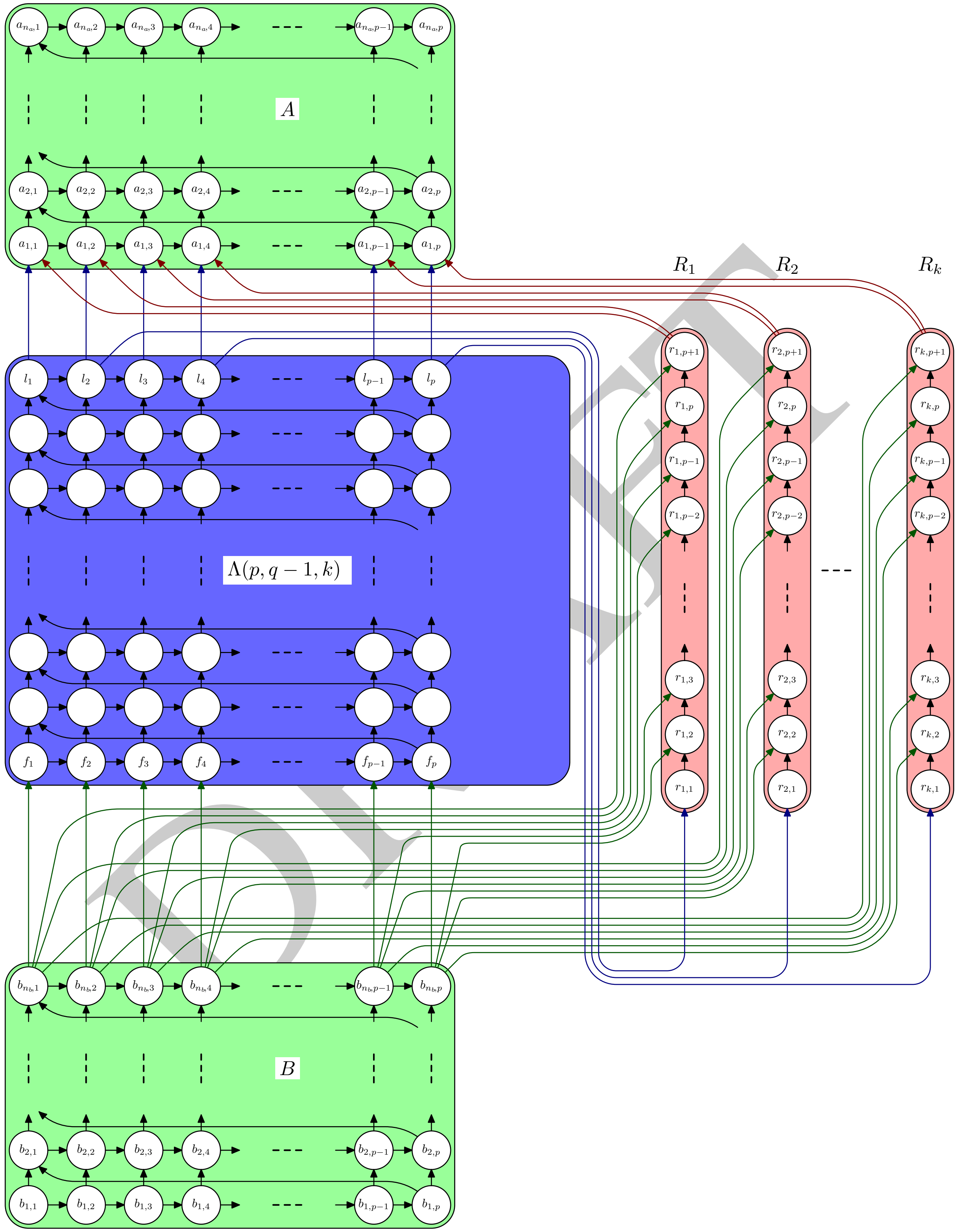


Figure 26: Building block $N(p, q, k)$ in polynomial-size graph $\Lambda(p, q, k)$ with $k = p/2$.

We once more leave it to the reader to verify the indegree and size bounds stated in Theorem 12.12 and concentrate on proving upper and lower bounds on pebbling price. Note that setting $k = 1$ in Definition 12.15 results in a graph that is fairly similar to the one in Definition 12.8, and indeed in this case we obtain Theorem 12.5 as a special case of Theorem 12.12 (or more precisely, we obtain it for the subrange of parameters that are the focus of our interest). As we will see below, however, the proof of the latter theorem is somewhat involved as compared to that of the former. We therefore chose to first present the easier proof in Section 12.1 in order to help the reader see what is going on in the more elaborate inductive step that will be needed below to establish Theorem 12.12.

Following the structure of the proof of Theorem 12.3, we first show an upper bound on the black-white pebbling price.

Lemma 12.16 ([KS91]). *BW-Peb*$^{\bullet}(K(p, q)) \leq p + 2kq + 1$.

Proof. The black-white pebbling strategy is very similar to that in the proof of Lemma 12.9. Again, the idea is that by keeping 2 pebbles each on all the graphs R_i, $i \in [k]$, at all levels of recursion in the graph construction, for a total of $2kq$ pebbles, we can use $p + 1$ black pebbles to do a simple bottom-up pebbling of the vertices in the column-part of the graph. For completeness, we give a brief description of the pebbling strategy below.

The strategy is constructed by induction over q. The base case is easy, since $\Lambda(p, 0, k)$ is just a p-line.

Inductively, suppose that we have constructed for $\Lambda(p, q - 1, k)$ a black-white pebbling $\mathcal{P}$ starting with black pebbles on the first row $(B_\sigma, W_\sigma) = (\{f_1, f_2, \ldots, f_p\}, \emptyset)$, ending with black pebbles on the last row $(B_\tau, W_\tau) = (\{l_1, l_2, \ldots, l_p\}, \emptyset)$, and never using more than $k(q - 1)$ white pebbles and $p + k(q - 1) + 1$ black pebbles. It is sufficient to construct from $\mathcal{P}$ a pebbling $\mathcal{P}'$ for the block graph $N(p, q, k)$ moving black pebbles from the first row of B to the last row of A using no more than kq white pebbles and $p+kq+1$ black pebbles. Such a pebbling is then easily extended to pebbling for all of $\Lambda(p, q, k)$ by pebbling the blocks one by one in a bottom-up fashion.

Thus, suppose that we have black pebbles on all vertices in the first row of B. Using one auxiliary black pebble, move all these black pebbles row by row, from left to right for each row, until the last row of B has all vertices covered by black pebbles.

Next, place white pebbles on all vertices $r_{i,1}$ and black pebbles on all vertices $r_{i,2}$ for $i \in [k]$, and then move the black pebbles all the way up to $r_{i,p+1}$, $i \in [k]$, with the help of one auxiliary black pebble. This is possible since all vertices in the last row of B have pebbles. At this point, we have a total of k black and k white pebbles on R_i, $i \in [k]$.

Now, shift the black pebbles from the last row of B to the first row of $\Lambda(p, q - 1, k)$, and appeal to the induction hypothesis to obtain a pebbling moving these black pebbles to the last row of $\Lambda(p, q - 1, k)$ using at most $p + k(q - 1) + 1$ black and $k(q - 1)$ white pebbles. We note that adding the k black and k white pebbles on the R-graphs, the total number of pebbles exactly meets the upper bound we are aiming for in the inductive step.

To finish the pebbling of $N(p, q, k)$, first remove all the white pebbles on $r_{i,1}$, $i \in [k]$, which is allowed since the predecessors of these vertices in the last row of $\Lambda(p, q - 1, k)$ are covered by pebbles. Then shift the black pebbles from the last row of $\Lambda(p, q - 1, k)$ to the first row of A, which is possible since the vertices $r_{i,p+1}$, $i \in [k]$, all have black pebbles, and remove the pebbles from these vertices $r_{i,p+1}$. Finally, move all the black pebbles in A row by row upwards, using one auxiliary black pebble, until the last row of A has all vertices covered by black pebbles. This concludes the inductive step, and the lemma follows. ☐

As in Section 12.1, some special notation for the first m blocks of $\Lambda(p, q, k)$ will come in handy in the lower bound proof.

Definition 12.17 (Subgraphs $\Lambda(p, q, k)[m]$ and $\Lambda^{+B}(p, q, k)[m]$ [KS91]). We let $\Lambda(p, q, k)[m]$ denote the subgraph of $\Lambda(p, q, k)$ consisting of the first m blocks $N^{(1)}(p, q, k), N^{(2)}(p, q, k), \ldots, N^{(m)}(p, q, k)$ and the edges between them. We write $\Lambda^{+B}(p,q,k)[m]$ to denote the subgraph $\Lambda(p,q,k)[m]$ extended by also including the subgraph $B^{(m+1)}$ of the $(m+1)$st block.

For the purpose of analysis in the proof, we will also partition the vertices in each block into so-called *slices*. See Figure 27 for an example of a slice in a N-block.

Definition 12.18 (Slice [KS91]). The ith *slice* of the mth N-block, denoted $\mathit{slice}(m,i)$, contains all vertices in $R_i^{(m)}$ as well as all columns j of $N^{(m)}(p,q,k)$ such that $(i-1)p/k < j \leq ip/k$.

With this notation, the black pebbling lower bound is proven inductively as follows.

Lemma 12.19 ([KS91]). *Suppose for $1 \leq q \leq p$ and $1 \leq m \leq \lceil p/k \rceil + 1$ that $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a black pebbling on $\Lambda(p,q,k)[m]$ such that some column c is completely free of pebbles at time σ but the last vertex $a_{n_a,c}^{(m)}$ in column c is pebbled at time τ. Then $\mathsf{space}(\mathcal{P}) \geq p(q-1) + k(m-1)$.*

In particular, any completely black pebbling strategy for $\Lambda(p,q,k)$ requires space at least pq.

The proof of Lemma 12.19 is similar in spirit to that of Lemma 12.11, but with some extra twists. As in Section 12.1, we start by giving an intuitive sketch of the argument before presenting the formal proof.

On a high level, in the proof of Lemma 12.11 we wanted to show that pebbling a column c spanning the $(m+1)$ first blocks required one more pebble than pebbling a column spanning just the m first blocks. The overall structure of the proof was that unless some special cases held, which were analyzed along the way and turned out to immediately yield the desired conclusion, we finally ended up in a case where we could prove that some column d spanning the first m blocks had to be pebbled during a time period where there was at least one pebble in the $(m+1)$st block. From this the induction step followed.

The inductive proof of Lemma 12.19 proceeds by analogous reasoning. Here, however, we need not only 1 pebble on the $(m+1)$st block but k pebbles for the induction step to go through. We therefore have to repeat the argument above recursively k times. For each recursive round, we prove that unless some special cases hold, thanks to which the induction step immediately follows, we can find some new column in the subgraph induced on the m first blocks and some slice in the $(m+1)$st block such that this slice, as well as all slices in the $(m+1)$st block found in previous rounds, must contain at least one pebble each while the column spanning the m first blocks is pebbled. Then we recurse again and start the proof all over from the beginning. After $k \leq p$ rounds of recursion, where we have to take some care to check that it is indeed possible to recurse p times (incidentally, this is the reason why there has to be so many rows in A), we get the required k pebbles, which establishes the lemma. The formal proof follows.

Proof of Lemma 12.19. By induction over q and m. The base case $q = m = 1$ is vacuously true.

First induction step: Our induction hypothesis is that the statement of the lemma holds true for $q' < q$ and also for $q' = q$ and $1 \leq m' \leq m < \lceil p/k \rceil + 1$. We show that this implies that it is true also for q and $m+1$, i.e., for the graph $\Lambda(p,q,k)[m+1]$.

Suppose $\mathcal{P} = \{\mathbb{P}_\sigma, \ldots, \mathbb{P}_\tau\}$ is a pebbling on $\Lambda(p,q,k)[m+1]$ as stated in the lemma. By assumption, column c in the subgraph $\Lambda(p,q,k)[m]$ is pebble-free at time σ but has been completely pebbled at time τ. Let $\sigma_1 + 1$ be the first time $b_{1,c}^{(1)}$ is pebbled and τ_1 be the first time $a_{n_a,c}^{(m)}$ is pebbled. We do the inductive step by a case analysis.

Case 1: Throughout the time interval $I_1 = [\sigma_1, \tau_1] \subseteq [\sigma, \tau]$, there is at least one pebble in each of the k slices of the $(m+1)$st block $N^{(m+1)}(p,q,k)$. If so, the induction step immediately follows.

Case 2: There is a time t_1 in $[\sigma_1, \tau_1]$ when some slice of the $(m+1)$st block, say $\mathit{slice}(m+1, d^*)$, is completely free of pebbles.

Consider the horizontal paths in B from column c in one row to column $c-1$ in the next, i.e., the paths along the vertices $b_{i,c}^{(m+1)}, b_{i,c+1}^{(m+1)}, \ldots, b_{i,p-1}^{(m+1)}, b_{i,p}^{(m+1)}, b_{i+1,1}^{(m+1)}, \ldots, b_{i+1,c-1}^{(m+1)}$ for $i = 1, \ldots, n_b - 1$. Without loss of generality we can assume that at least one of these paths is pebble-free at time t_1, say the path Q_B starting at row i_b, since otherwise $\mathit{space}(\mathbb{P}_{t_1}) \geq n_b - 1 > p^2 \geq pq$ and we are done. For the same reason, there is at least one pebble-free path $a^{(m+1)}_{i,c}, a^{(m+1)}_{i,c+1}, \ldots, a^{(m+1)}_{i,p}, a^{(m+1)}_{i+1,1}, \ldots, a^{(m+1)}_{i+1,c-1}$ in A for some row $i = n_a - (p^2+1), n_a - p^2, \ldots, n_a - 2$. Let us fix such a path, say the path Q_A starting at row i_a.

This implies that at time t_1 we can find a pebble-free path Q in $N^{(m+1)}(p,q,k)$ from $b^{(m+1)}_{1,c}$ to $a^{(m+1)}_{n_a,c}$ in the following way:

- Go from $b^{(m+1)}_{1,c}$ along column c to row i_b in B.
- Follow Q_B from column c to column $d = \lfloor d^* p/k \rfloor$, which is in $\mathit{slice}(m+1, d^*)$.
- Go along column d through $\Lambda^{(m+1)}(p, q-1, k)$, entering at $f^{(m+1)}_d$ and exiting at $l^{(m+1)}_d$, all the way up to row i_a in A.
- Follow Q_A from column d back to column c.
- Continue up along column c to $a^{(m+1)}_{n_a,c}$.

See Figure 27 for a schematic illustration of what such a path might look like.

The path Q is thus empty at time t_1 but must be have been completely pebbled by time τ. Column c in the $(m+1)$st N-block can impossibly be pebbled before time τ_1, however (when a pebble is placed on $a^{(m)}_{n_a,c}$, the last vertex in column c in the mth block), so the whole path Q remains pebble-free from time t_1 until time τ_1.

Let $I_2 = [\sigma_2, \tau_2]$ be a time interval such that $\sigma_2 \geq \tau_1$, a pebble is placed on the vertex in column d of Q_B at time $\sigma_2 + 1$, the vertex in column d of Q_A is pebbled for the first time after τ_1 at time τ_2, and σ_2 is minimal such that $\mathit{slice}(m+1, d^*)$ contains at least one pebble throughout $(\sigma_2, \tau_2]$ (such a time interval must exist by the frugality of the pebbling; see Definition 3.12 and Lemma 3.13). Note that $\tau_2 < \tau$ since Q must have been completely pebbled by time τ.

Consider the subinterval $I_3 = [\sigma_3, \tau_3] \subseteq I_2$ such that $f^{(m+1)}_d$ is pebbled for the first time after σ_2 at time $\sigma_3 + 1$ and $l^{(m+1)}_d$, is pebbled for the first time after σ_2 at time τ_3. Observe that by the minimality of σ_2, column d in $\Lambda^{(m+1)}(p, q-1, k)$ must be empty of pebbles at time σ_3. Recall that $\Lambda^{+B}(p,q,k)[m]$ denotes the subgraph of $\Lambda(p,q,k)[m+1]$ induced on the vertices of $\Lambda(p,q,k)[m]$ plus $B^{(m+1)}$. We have two cases.

Case 2a: Throughout $[\sigma_3, \tau_3]$ there is at least one pebble on each of the p columns of $\Lambda^{+B}(p,q,k)[m]$. If so, the induction step follows for $q > 1$ by appealing to the induction hypothesis for $\Lambda(p, q-1, k)$, since we get a total of at least $p + p(q-2) + k\lceil p/k \rceil \geq p(q-1) + km$ pebbles. For $q = 1$, we instead get the desired bound by observing directly that $p \geq mk$.

Case 2b: There is a time $t_3 \in [\sigma_3, \tau_3]$ when some column in $\Lambda^{+B}(p,q,k)[m]$, say column e, is completely pebble-free. Let us pause and collect what we know so far:

- By the minimality of σ_2, at time σ_2 column d in $\Lambda(p, q-1, k)$ as well as all of $R^{(m+1)}_{d^*}$ is empty.
- The vertex $l^{(m+1)}_d$ is not pebbled after σ_2 until time τ_3, which means that $R^{(m+1)}_{d^*}$ must still be pebble-free at time τ_3.
- The vertex in column d of Q_A—let us denote it $a^{(m+1)}_{n',d}$—gets a pebble at time $\tau_2 > \tau_3$, where the inequality holds since the vertex $l^{(m+1)}_d$, which receives a pebble at time τ_3, is an ancestor of this vertex.
- Since the vertices in $R^{(m+1)}_{d^*}$ are ancestors of $a^{(m+1)}_{n',d}$, all of the subgraph $R^{(m+1)}_{d^*}$ must be completely pebbled during $[\tau_3, \tau_2]$.

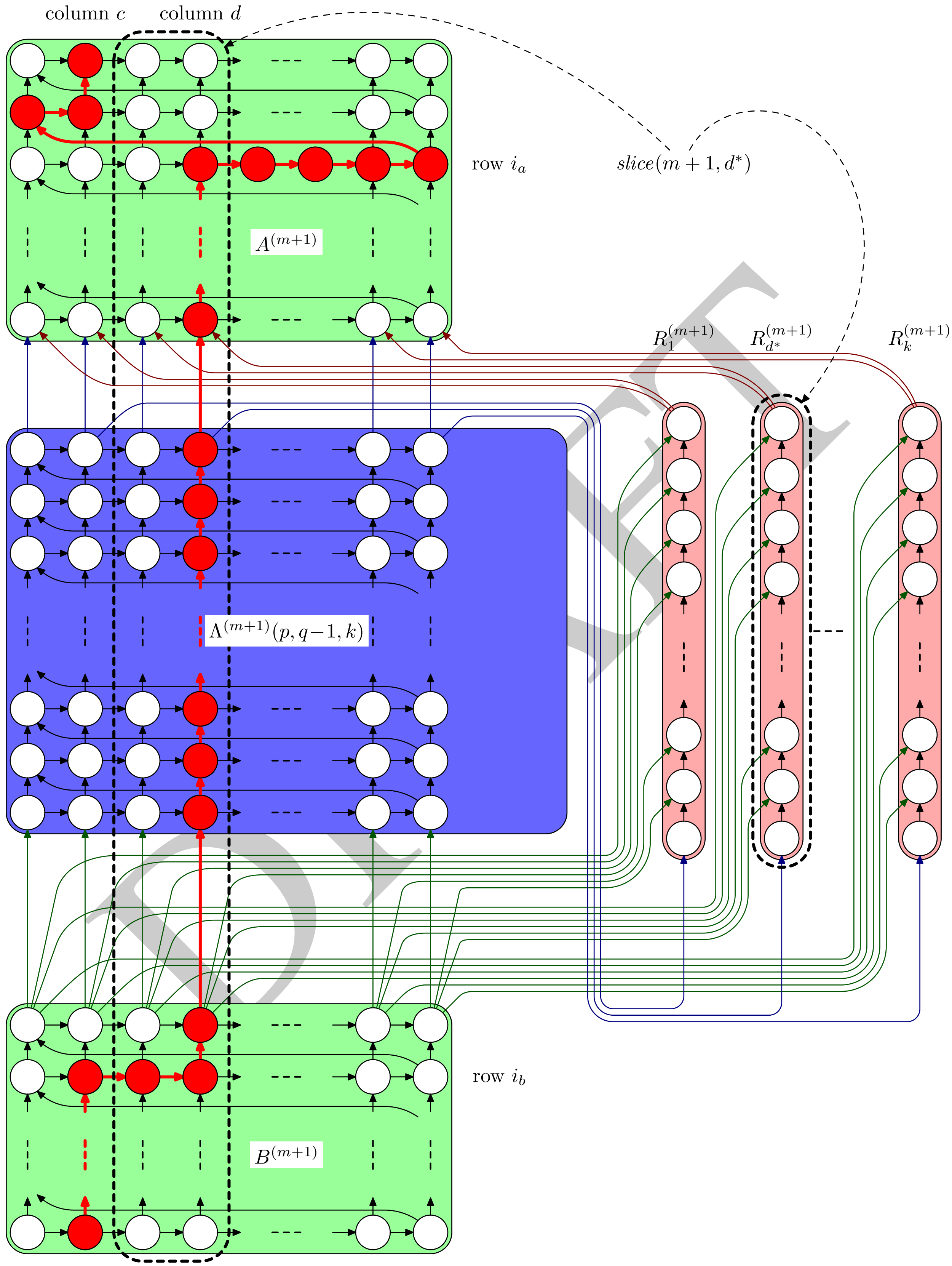


Figure 27: Path Q in $N^{(m+1)}(p,q,k)$ from $b_{1,c}^{(m+1)}$ to $a_{n_a,c}^{(m+1)}$ via d^*th slice for $c = 2$, $d = 4$, and $d^* = 2$.

- But the vertices in column e of $\Lambda^{+B}(p,q,k)[m]$ are ancestors of $R^{(m+1)}_{d^*}$ so this column must be pebbled after time t_3 and before time τ_2.

- Hence, in particular, the eth column spanning $\Lambda(p,q,k)[m]$ is completely pebbled during (t_3, τ_2), and by construction there is at least one pebble on $slice(m+1, d^*)$ throughout this interval.

That is, we have shown that there is a subinterval of $[\sigma_1, \tau_1]$ when some column in $\Lambda(p,q,k)[m]$ is completely pebbled while some slice in the $(m+1)$st block contains at least one pebble.

This sets the stage for rewinding the proof and repeating the whole inductive argument from the beginning. Some modifications are needed, though, since we can no longer without loss of generality make assumptions about the p^2+2 last rows of A in case 2. Therefore, we do not consider these rows and instead of $a^{(m+1)}_{n_a,c}$ we let the goal vertex of the new pebbling under consideration be the vertex in the dth column of our previously constructed path Q_A, i.e., the vertex $a^{(m+1)}_{n',d}$. Note that $n' \geq n_a - (p^2+1)$.

Before $a^{(m+1)}_{n',d}$ is pebbled, the column e spanning $\Lambda(p,q,k)[m]$ must be completely pebbled. Let $I'_1 = [\sigma'_1, \tau'_1]$ be the interval such that $b^{(1)}_{1,e}$ is pebbled for the first time after t_3 at time σ'_1 and $a^{(m)}_{n_a,e}$ is pebbled for the first time after t_3 at time τ'_1. Recall that by the analysis above, we have $\tau'_1 \leq \tau_2$. We know that at least one slice in the $(m+1)$st block contains a pebble throughout I'_1. If all slices do, then we are done as in case 1 above.

Otherwise, we repeat the argument in case 2 and find a pebble-free path Q' in $N^{(m+1)}(p,q-1,k)$ from $b^{(m+1)}_{1,e}$ via $slice(m+1, f^*)$ to some column f and then onwards to column d in A reaching $a^{(m+1)}_{n',d}$. To be guaranteed to find a pebble-free subpath through A in this step, this time we look at the rows $n_a - (2p^2+1), n_a - 2p^2, \ldots, n_a - (p^2+2)$. Note also that nothing is really changed in the reasoning by the fact that column e in B is not the same column as column d in A. Therefore, the whole argument goes through again.

To clinch the proof of the inductive step, observe that we need only recurse $k \leq p$ times. After that, we are guaranteed to have pebbles in all slices of the $(m+1)$st block and case 1 applies. And we can apply the recursive argument p times, since A contains $n_a > 1 + (p+1)(p^2+1)$ rows.

Second induction step: Suppose that the lemma holds for all $q' \leq q$ and all $m' \leq \lceil p/k \rceil + 1$. To establish the lemma for $q' = q+1$ and $m' = 1$, just note that $\Lambda(p,q+1,k)[1]$ contains a copy of $\Lambda(p,q,k)$, which by the induction hypothesis requires $p(q-1) + k\lceil p/k \rceil \geq pq$ pebbles. □

Combining Lemmas 12.16 and 12.19, Theorem 12.12 follows.

13 Some Pebbling Results Not Covered in This Survey

In this section we briefly mention some pebbling results that are not covered in detail in this survey. This list of omissions is non-exhaustive.

13.1 Sharp Time-Space Trade-offs

Some pebbling papers have studied "sharp" or "abrupt" time-space trade-offs, where the pebbling time can increase dramatically for just a small decrease of the pebbling space. Such results are somewhat orthogonal to the main concerns of this survey, where we have focused on trade-offs that have a certain "robustness" property in that they do not depend crucially on small changes of the multiplicative or additive constants involved.

Some references for this kind of sharp trade-off results are [PT78], which presents an exponential time increase when removing a constant fraction of pebbles, [Lin78], which established an exponential increase when removing just two pebbles, and [EBL79, GLT80, Sav98], which obtains an exponential jump in time when removing just one pebble.

These results are all for black pebbling. For black-white pebbling we are only aware of [HP10], which gets a sharp exponential trade-off when decreasing the space by one. As opposed to other pebbling results that hold for bounded fan-in DAGs, however, this black-white exponential trade-off requires graphs with unbounded indegree.

13.2 Complexity of Determining Pebbling Price

Some of the sharp time-space trade-offs discussed above have been related to work on trying to determine the complexity of deciding the pebbling price of a given DAG G. Formally, this is the problem of, given a DAG G and a space limit s, answering the question whether there is a complete pebbling strategy of G in space at most s or not.

This decision problem is easily seen to be in PSPACE for all flavours of pebbling, since using that PSPACE = NPSPACE we can nondeterministically guess a pebbling strategy in space at most s and keep a counter so that we abort when the pebbling has gone on for time longer than the total number of distinct pebble configurations on the graph. Sethi [Set75] showed that the special case for black pebblings of a DAG G when we allow every vertex to be pebbled only once is NP-complete. Lingas [Lin78] was able to prove PSPACE-completeness of a generalized pebble game, where the DAG has both AND- and OR-nodes (the standard pebble game corresponding to AND-nodes only) and where the pebbling rules are modified accordingly, and this was extended by Gilbert, Lengauer, and Tarjan [GLT80] who proved that the general decision problem for standard black pebbling is in fact PSPACE-complete. This result was considered somewhat surprising at the time since pebbling is a one-player game without the kind of alternation between two players found in other PSPACE-completeness result.

Theorem 13.1 ([GLT80]). *The decision problem whether a single-sink DAG G with fan-in at most 2 can be black-pebbled with at most s pebbles is* PSPACE*-complete.*

An obvious question is whether the same PSPACE-completeness result holds also for the black-white pebble game. This was mention as an "open problem of the month" in David Johnson's *NP-Completeness Column* [Joh83]. If every vertex can be pebbled exactly once we again have NP-completeness as shown by Lengauer [Len81], but the general case has remained completely open until the recent work of Hertel and Pitassi [HP10], which establish PSPACE-completeness provided that we allow unbounded fan-in.

Theorem 13.2 ([HP10]). *The decision problem whether a single-sink DAG G of unbounded fan-in has a complete black-white pebbling in space at most s is* PSPACE*-complete.*

Presumably, Theorem 13.2 should hold for graphs with bounded indegree as well, but this is open. Unfortunately, it appears that the construction in Theorem 13.2 depends crucially on the fact that the graphs have unbounded fan-in.

Open Problem 9. *Is the problem of determining the black-white pebbling price of graphs with constant indegree* PSPACE*-complete?*

One reason for presenting the exact bounds on black-white pebbling price of pyramids in Section 4 is that such graphs are an important building block in the construction in Theorem 13.1. One prerequisite for lifting this construction to black-white pebbling would therefore seem to be the knowledge of the exact black-white pebbling price of pyramids.

13.3 Extensions of The Pebble Game

This survey has focused mostly on the extension of the black pebble game to black-white pebbling, which is a way of modelling nondeterministic computation. In addition to the black-white pebble game, a number of other variants of pebbling have been introduced, such as the pebble game with auxiliary pushdowns, the red-blue pebble game, and the memory hierarchy game. We make no attempt to cover these extensions in the current paper, but instead refer to [Pip80, Sav98] for more details.

14 Pebbling and Proof Complexity

In this section,[4] we describe the connections between pebbling and proof complexity that is the main motivation behind this survey. Our focus will be on how pebble games have been employed to study trade-offs between time and space in resolution-based proof systems, in particular in the line of work [Nor09a, NH08b, BN08, BN11], but we will also discuss other usages of pebbling in proof complexity. Let us start, however, by giving a quick overview of proof complexity in general.

14.1 A Selective Introduction to Proof Complexity

The study of proof complexity originated with the seminal paper of Cook and Reckhow [CR79]. In its most general form, a proof system for a language L is a predicate $P(x, \pi)$, computable (deterministically) in time polynomial in the sizes $|x|$ and $|\pi|$ of the input, and having the property that for all $x \in L$ there is a string π (a *proof*) for which $P(x, \pi)$ evaluates to true, whereas for any $x \notin L$ it holds for all strings π that $P(x, \pi)$ evaluates to false. A proof system is said to be polynomially bounded if for every $x \in L$ there exists a proof π_x for x that has 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 [CR79] 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 propositional proof systems, it seems that we still do not fully understand the reasoning power of even quite simple ones.

Another important motivation for proof complexity is that 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. In the last 10-15 years, SATISFIABILITY has gone from a problem of mainly theoretical interest to a practical approach for solving applied problems. Although all known Boolean satisfiability solvers (SAT solvers) have exponential running time in the worst case, enormous progress in performance has led to satisfiability algorithms becoming a standard tool for solving a large number of real-world problems such as hardware and software verification, experiment design, circuit diagnosis, and scheduling (see, for instance, [KS07, Mar08] for more details).

All SAT solvers, regardless of whether they produce a written proof or not, explicitly or implicitly define a system in which proofs are searched for and rules which determine what proofs in this system look like. Proof complexity analyzes what it takes to simply write down and verify the proofs that such an automated

[4] This section is adapted and abbreviated from the paper [Nor10b].

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.

The field of proof complexity also has rich connections to cryptography, artificial intelligence and mathematical logic. We again refer the reader to [Bea04, BP98, CK02, Seg07, Urq95] for more information.

14.1.1 Resolution-Based Proof Systems

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 that produces refutations of unsatisfiable CNF formulas can be considered as a general propositional proof system.

Arguably the single most studied proof system in propositional proof complexity, *resolution*, is such a system for deriving proofs of the unsatisfiability of CNF formulas. The resolution proof system appeared in [Bla37] and began to be investigated in connection with automated theorem proving in the 1960s [DLL62, DP60, Rob65]. Because of its simplicity—there is only one derivation rule—and because all lines in a proof are disjunctive clauses, this proof system readily lends itself to proof search algorithms. In fact, the most successful SAT solvers to date, as witnessed by recent rounds of the international SAT competition [SAT], are based on the so-called Davis-Putnam-Logemann-Loveland or DPLL procedure [DLL62, DP60] which produces resolution proofs.

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. The first superpolynomial lower bound on resolution was presented by Tseitin in the paper [Tse68] for a restricted form of the proof system called *regular* resolution. It took almost an additional 20 years before Haken [Hak85] was able to establish superpolynomial bounds without any restrictions, showing that CNF encodings of the pigeonhole principle are intractable for general resolution. This weakly exponential bound of Haken has later been followed by many other strong results, among others truly exponential lower bounds on resolution refutation length for different formula families in, for instance, [BKPS02, BW01, CS88, Urq87].

A second complexity measure for resolution proofs is the *width*, measured as the maximal size of a clause in the proof. Clearly, the maximal width needed to refute an unsatisfiable CNF formula is the number of variables in it, which is upper-bounded by the formula size. Hence, while the refutation length can be exponential in the worst case, the width ranges between constant and linear measured in the formula size. Inspired by previous work [BP96, CEI96], Ben-Sasson and Wigderson [BW01] identified width as a crucial resource of resolution proofs by showing that the minimal width of any resolution refutation of a k-CNF formula F (i.e., a formula where all clauses have size at most some constant k) is bounded from above by the minimal refutation length by

$$\text{minimal width} \leq \mathrm{O}\Big(\sqrt{(\text{size of formula}) \cdot \log(\text{minimal length})}\Big). \tag{14.1}$$

Since it is also easy to see that resolution refutations of polynomial-size formulas in small width must necessarily be short—quite simply for the reason that $(2 \cdot \#\text{variables})^w$ is an upper bound on the total number of distinct clauses of width w—the result in [BW01] 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.

The study of space as a resource for resolution was initiated by Esteban and Torán in [ET01, Tor99] and was later extended to a more general setting by Alekhnovich et al. in [ABRW02]. Intuitively, we can view a resolution refutation of a CNF formula F as a sequence of derivation steps on a blackboard, where in each step we may write a clause from F on the blackboard, erase a clause from the blackboard, or derive some new clause implied by the clauses currently written on the blackboard, and where the refutation ends when we reach the contradictory empty clause. The space of a refutation is then the maximal number of clauses one needs to keep on the blackboard simultaneously at any time during the refutation, and the space of refuting F is defined as the minimal space of any resolution refutation of F. A number of upper and lower bounds for refutation space in resolution and other proof systems were subsequently presented in, for example, [ABRW02, BG03, EGM04, ET03], and to distinguish the space measure of [ET01, Tor99] from other measures introduced in these papers we will sometimes refer to it as *clause space* below for extra clarity.

Just as is the case for width, the minimum clause space of refuting a formula can be upper-bounded by the formula size. Somewhat unexpectedly, it was discovered in a sequence of works that lower bounds on resolution refutation space for different formula families turned out to match exactly previously known lower bounds on refutation width. In an elegant paper [AD08], Atserias and Dalmau showed that this was not a coincidence, but that the inequality

$$\text{minimal width} \leq \text{minimal clause space} + \text{small constant} \tag{14.2}$$

holds for refutations of any k-CNF formula F, where the constant term depends only on k. Combining the inequality in (14.2) with the counting argument for width versus length mentioned above, it follows that upper bounds on clause space imply upper bounds on length. Esteban and Torán [ET01] showed the converse that length upper bounds imply clause space upper bounds for the restricted case of *tree-like* resolution (where every clause can only be used once in the derivation and has to be rederived again from scratch if it is needed again at some later stage in the proof). Thus, clause space is an interesting complexity measure with nontrivial relations to proof length and width. We note that apart from being of theoretical interest, clause space has also been proposed in [ABLM08] as an adequate measure of the hardness in practice of CNF formulas for DPLL-based SAT solvers.

The resolution proof system was extended by Krajíček [Kra01] to the family of *k-DNF resolution* proof systems forming an intermediate step between resolution and depth-2 Frege systems. Roughly speaking, for positive integers k the kth member of this family, which we denote $\mathcal{R}(k)$, is allowed to reason in terms of formulas in disjunctive normal form (*DNF formulas*) with the added restriction that any conjunction in any formula is over at most k literals. For $k = 1$, the lines in the proof are hence disjunctions of literals, and the system $\mathcal{R}(1) = \mathcal{R}$ is standard resolution. At the other extreme, $\mathcal{R}(\infty)$ is equivalent to depth-2 Frege.

The original motivation to study this family of proof systems, as stated in [Kra01], was to better understand the complexity of counting in weak models of bounded arithmetic, and it was later observed that these systems are also related to SAT solvers that reason using multi-valued logic (see [JN02] for a discussion of this point). A number of subsequent works have shown superpolynomial lower bounds on the length of $\mathcal{R}(k)$-refutations, most notably for (various formulations of) the pigeonhole principle and for random CNF formulas [AB04, ABE02, Ale05, JN02, Raz03, SBI04, Seg05]. Of particular interest in the current context are the results of Segerlind et al. [SBI04] and of Segerlind [Seg05] showing that the family of $\mathcal{R}(k)$-systems form a strict hierarchy with respect to proof length. More precisely, they prove that for every k there exists a family of formulas $\{F_n\}_{n=1}^{\infty}$ of arbitrarily large size n such that F_n has a $\mathcal{R}(k+1)$-refutation of polynomial length but all $\mathcal{R}(k)$-refutations of F_n require exponential length.

With regard to space, Esteban et al. [EGM04] established essentially optimal linear lower bounds in $\mathcal{R}(k)$ on *formula space*, extending the clause space measure for resolution in the natural way by counting the number of k-DNF formulas. They also proved that the family of *tree-like* $\mathcal{R}(k)$ systems form a strict hierarchy with respect to formula space in the sense that there are arbitrarily large formulas F_n of size n that can be refuted in tree-like $\mathcal{R}(k+1)$ in constant space but require space $\Omega(n/\log^2 n)$ to be refuted in tree-like $\mathcal{R}(k)$. It should be pointed out, however, that as observed in [Kra01, EGM04] the family of tree-like $\mathcal{R}(k)$ systems for all $k > 0$ are strictly weaker than standard resolution. As was mentioned above, the family of general, unrestricted $\mathcal{R}(k)$ systems are strictly stronger than resolution, so the results in [EGM04] left completely open the question of whether there is a strict formula space hierarchy for (non-tree-like) $\mathcal{R}(k)$ or not.

14.1.2 Three Questions Regarding Space

Although resolution is simple and by now very well-studied, the research surveyed above left open a few fundamental questions about this proof system. In what follows, our main focus will be on the three questions considered below.[5]

1. What is the relation between clause space and width? The inequality (14.2) says that clause space $\gtrapprox$ width, but it leaves open whether this relationship is tight or not. Do the clause space and width measures always coincide, or is there a formula family that separates the two measures asymptotically?

2. What is the relation between clause space and length? Is there some nontrivial correlation between the two in the sense that formulas refutable in short length must also be refutable in small space, or can "easy" formulas with respect to length be "arbitrarily complex" with respect to space? (We will make these notions more precise shortly.)

3. Can the length and space of refutations be optimized simultaneously, or are there trade-off between the two measures in the sense that there are formulas for which any refutation, as soon as it gets anywhere close to using the minimal amount of space, must exhibit polynomial or even exponential blow-up in length?

To put the questions about length versus space in perspective, consider what has been known for length versus width. It follows from the inequality (14.1) that if the width of refuting a k-CNF formula family $\{F_n\}_{n=1}^{\infty}$ of size n grows asymptotically faster than $\sqrt{n \log n}$, then the length of refuting F_n must be superpolynomial. This is known to be almost tight, since Bonet and Galesi [BG01] showed that there is a family of k-CNF formulas of size n with minimal refutation width growing like $\sqrt[3]{n}$, but which is nevertheless refutable in length linear in n. Hence, formulas refutable in polynomial length can have somewhat wide minimum-width refutations, but not arbitrarily wide ones.

Turning to the relation between clause space and length, we note that the inequality (14.2) tells us that any correlation between length and clause space cannot be tighter than the correlation between length and width. 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 of F in length L it is a straightforward consequence of [ET01, HPV77] that the space needed to refute F is at most the order of $L/\log L$. This provdes an upper bound on any possible separation of the two measures. Thus, what the question above is asking is whether it can be that length and space are "completely unrelated" in the sense that there exist k-CNF formulas with refutation length L that need maximum possible space $\Omega\bigl(L/\log L\bigr)$, or whether there is a nontrivial upper bound on clause space in terms of length similar to the inequality in (14.1), say minimal clause space $\leq \mathrm{O}\bigl(\sqrt{(\text{size of formula}) \cdot \log(\text{minimal length.})}\bigr)$ or so. Intriguingly, as we discussed above it was shown in [ET01] that for the restricted case of so-called tree-like resolution there is in fact a tight correspondence between length and clause space, exactly as for length versus width.

[5] In the interest of full disclosure, it should perhaps be noted that these questions also happened to be the focus of the author's PhD thesis [Nor08].

14.1.3 Pebble Games to the Rescue

Although the above questions have been around for a while, as witnessed by discussions in, for instance, the papers [ABRW02, Ben09, EGM04, ET03, Seg07, Tor04], there appears to have been no consensus on what the right answers should be. However, what most of these papers did agree on was that a plausible formula family for answering these questions were so-called *pebbling contradictions*, which are CNF formulas encoding pebble games played on graphs. Pebbling contradictions had already appeared in various disguises in some of the papers mentioned in Section 14.1.1, and it had been noted that non-constant lower bounds on the clause space of refuting pebbling contradictions would separate space and width and possibly also clarify the relation between space and length if the bounds were good enough. On the other hand, a constant upper bound on the refutation space would improve the trade-off results for different proof complexity measures for resolution in [Ben09].

And indeed, pebbling turned out to be just the right tool to understand the interplay of length and space in resolution. The main purpose of this section is to give an overview of the works establishing connections between pebbling and proof complexity with respect to time-space trade-offs. We will need to give some preliminaries in order to state the formal theorems, but before we do so let us conclude this introduction by giving a brief description of the relevant results.

The first progress was reported in 2006 (journal version in [Nor09a]), where pebbling formulas of a very particular form, namely pebbling contradictions defined over complete binary trees, were studied. This was sufficient to establish a logarithmic separation of clause space and width, thus answering question 1 above. This separation was later improved from logarithmic to polynomial in [NH08b], where a broader class of graphs were analyzed, but where unfortunately a rather involved argument was required for this analysis to go through. In [BN08], a somewhat different approach was taken by slightly modifying the pebbling formulas. This made the analysis both much simpler and much stronger, and led to a resolution of question 2 by establishing an optimal separation between clause space and length, i.e., that there are formulas with refutation length L that require clause space $\Omega\big(L/\log L\big)$. In a further improvement, [BN11] used similar ideas to translate pebbling time-space trade-offs to trade-offs between length and space in resolution, thus answering question 3. The paper [BN11] also extended these results to the k-DNF resolution proof systems, establishing as a corollary that the $\mathcal{R}(k)$-systems indeed form a strict hierarchy with respect to space.

14.2 Proof Complexity Preliminaries

Below we present the definitions, notation and terminology that we will need to make more precise the informal exposition in Section 14.1.

14.2.1 Variables, Literals, Terms, Clauses, Formulas and Truth Value Assignments

For x a Boolean variable, a *literal over* x is either the variable x itself, called a *positive literal over* x, or its negation, denoted $\neg x$ or $\overline{x}$ and called a *negative literal over* x. Sometimes it will also be convenient to write x^1 for unnegated variables and x^0 for negated ones. We define $\neg\neg x$ to be x. A *clause* $C = a_1 \vee \cdots \vee a_k$ is a disjunction of literals, and a *term* $T = a_1 \wedge \cdots \wedge a_k$ is a conjunction of literals. Below we will think of clauses and terms as sets, so that the ordering of the literals is inconsequential and that, in particular, no literals are repeated. We will also (without loss of generality) assume that all clauses and terms are nontrivial in the sense that they do not contain both a literal and its complement. A clause (term) containing at most k literals is called a *k-clause* (*k-term*). A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a conjunction of clauses, and a *DNF formula* is a disjunction of terms. We will think of CNF and DNF formulas as sets of clauses and terms, respectively. A *k-CNF formula* is a CNF formula consisting of k-clauses, and a *k-DNF formula* consists of k-terms.

The *variable set* of a term T, denoted $\mathit{Vars}(T)$, is the set of Boolean variables over which there are literals in T, and $\mathit{Lit}(T)$ is the set of literals. The variable and literal sets of a clause are similarly defined and these definitions are extended to CNF and DNF formulas by taking unions.[6] If V is a set of Boolean variables and $\mathit{Vars}(T) \subseteq V$ we say T is a term *over* V and similarly define clauses, CNF formulas, and DNF formulas over V.

In what follows, we will usually write a, b, c to denote literals, A, B, C, D to denote clauses, T to denote terms, F, G to denote CNF formulas, and $\mathbb{C}, \mathbb{D}$ to denote sets of clauses, k-DNF formulas or sometimes other Boolean functions. We will be working with an arbitrary but fixed set of variables $V = \{x, y, z, \ldots\}$. For a variable $x \in V$ we define $\mathit{Vars}^d(x) = \{x_1, \ldots, x_d\}$, and we will tacitly assume that V is such that the set of variables $\mathit{Vars}^d(V) = \{x_1, \ldots, x_d, y_1, \ldots, y_d, z_1, \ldots, z_d, \ldots\}$ is disjoint from V. We will say that the variables $x_1, \ldots, x_d$, and any literals over these variables, all *belong* to the variable x.

We write α, β to denote truth value assignments, usually over $\mathit{Vars}^d(V)$ but sometimes over V. Partial truth value assignments, or *restrictions*, will often be denoted ρ. Truth value assignments are functions to $\{0, 1\}$, where we identify 0 with false and 1 with true. We have the usual semantics that a clause is true under α, or *satisfied* by α, if at least one literal in it is true, and a term is true if all literals evaluate to true. We write 0 to denote the empty clause without literals that is false under all truth value assignments. (The empty clause is also denoted λ or Λ in the literature.) A CNF formula is satisfied if all clauses in it are satisfied, and for a DNF formula we require that some term should be satisfied. In general, we will not distinguish between a formula and the Boolean function computed by it.

If $\mathbb{C}$ is a set of Boolean functions we say that a restriction (or assignment) satisfies $\mathbb{C}$ if and only if it satisfies every function in $\mathbb{C}$. For $\mathbb{D}, \mathbb{C}$ two sets of Boolean functions over a set of variables V, we say that $\mathbb{D}$ *implies* $\mathbb{C}$, denoted $\mathbb{D} \vDash \mathbb{C}$, if and only if every assignment $\alpha : V \mapsto \{0, 1\}$ that satisfies $\mathbb{D}$ also satisfies $\mathbb{C}$. In particular, $\mathbb{D} \vDash 0$ if and only if $\mathbb{D}$ is *unsatisfiable* or *contradictory*, i.e., if no assignment satisfies $\mathbb{D}$. If a CNF formula F is unsatisfiable but for any clause $C \in F$ it holds that the clause set $F \setminus \{C\}$ is satisfiable, we say that F is *minimally unsatisfiable*.

14.2.2 Proof Systems

In this paper, we will focus on proof systems for refuting unsatisfiable CNF formulas. (As was discussed in Section 14.1.1 this is essentially without loss of generality.) In this context it should be noted that, perhaps somewhat confusingly, a refutation of a formula F is often also referred to as a "proof of F" in the literature. We will try to be consistent and talk only about "refutations of F," but will otherwise use the two terms "proof" and "refutation" interchangeably.

We say that a proof system $\mathcal{P}$ is *sequential* if a proof π in $\mathcal{P}$ is a *sequence* of lines $\pi = \{L_1, \ldots, L_\tau\}$ of some prescribed syntactic form depending on the proof system in question, where each line is derived from previous lines by one of a finite set of allowed *inference rules*. Following the exposition in [ET01], we view a proof as similar to a Turing machine computation, with a special read-only input tape from which the clauses of the formula F being refuted (the *axioms*) can be downloaded and a working memory where all derivation steps are made. Then the length of a proof is essentially the time of the computation and space measures memory consumption. The following definition is the straightforward generalization of the setup in [ABRW02] to arbitrary sequential proof systems. We note that proofs defined in this way have been referred to as *configuration-style* proofs or *space-oriented* proofs in the literature.

Definition 14.1 (Refutation). For a sequential proof system $\mathcal{P}$ with lines of the form L_i, a $\mathcal{P}$-*configuration* $\mathbb{D}$, or, simply, a *configuration*, is a set of such lines. A sequence of configurations $\{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ is

[6] Although the notation $\mathit{Lit}(\cdot)$ is slightly redundant for clauses and terms given that they are defined as sets of literals, we find that it increases clarity to have a uniform notation for literals appearing in clauses or terms *or formulas*. Note that $x \in F$ means that that the unit clause x appears in the CNF formula F, whereas $x \in \mathit{Lit}(F)$ denotes that the positive literal x appears in some clause in F and $x \in \mathit{Vars}(F)$ denotes that the variable x appears in F with unknown sign.

said to be a $\mathcal{P}$*-derivation* from a CNF formula F if $\mathbb{D} = \emptyset$ and for all $t \in [\tau]$, the set $\mathbb{D}_t$ is obtained from $\mathbb{D}_{t-1}$ by one of the following *derivation steps*:

Axiom Download $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L_C\}$, where L_C is the representation of some clause $C \in F$ (an *axiom clause*).

Inference $\mathbb{D}_t = \mathbb{D}_{t-1} \cup \{L\}$ for some L inferred by one of the inference rules for $\mathcal{P}$ from a set of assumptions $L_1, \ldots, L_m \in \mathbb{D}_{t-1}$.

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

A $\mathcal{P}$-refutation $\pi : F \vdash 0$ of a CNF formula F is a derivation $\pi = \{\mathbb{D}_0, \ldots, \mathbb{D}_\tau\}$ such that $\mathbb{D}_0 = \emptyset$ and $0 \in \mathbb{D}_\tau$, where 0 is the representation of contradiction (e.g. for resolution and $\mathcal{R}(k)$-systems the empty clause without literals).

If every line L in a derivation is used at most once before being erased (though it can possibly be rederived later), we say that the derivation is *tree-like*. This corresponds to changing the inference rule so that $L_1, \ldots, L_d$ must all be erased after they have been used to derive L.

To every refutation π we can associate a DAG G_π, with the lines in π labelling the vertices and with edges from the assumptions to the consequence for each application of an inference rule. There might be several different derivations of a line L during the course of the refutation π, but if so we can label each occurrence of L with a time-stamp when it was derived and keep track of which copy of L is used where. Using this representation, a refutation π can be seen to be tree-like if G_π is a tree.

Definition 14.2 (Refutation size, length and space). Given a size measure $S(L)$ for lines L in $\mathcal{P}$-proofs (which we usually think of as the number of symbols in L, but other definitions can also be appropriate depending on the context), the *size* of a $\mathcal{P}$-proof π is the sum of the sizes of all lines in a proof, where lines that appear multiple times are counted with repetitions (once for every vertex in G_π). The *length* of a $\mathcal{P}$-proof π is the number of axiom downloads and inference steps in it, i.e., the number of vertices in G_π. For a space measure $Sp_{\mathcal{P}}(\mathbb{D})$ defined for $\mathcal{P}$-configurations, the *space* of a proof π is defined as the maximal space of a configuration in π.

If π is a refutation of a formula F in size S and space s, then we say that F can be refuted in size S and space s *simultaneously*. Similarly, F can be refuted in length L and space s simultaneously if there is a $\mathcal{P}$-refutation $\mathcal{P}$ with $L(\pi) = L$ and $Sp(\pi) = s$.

We define the $\mathcal{P}$*-refutation size* of a formula F, denoted $S_{\mathcal{P}}(F \vdash 0)$, to be the minimum size of any $\mathcal{P}$-refutation of it. The $\mathcal{P}$*-refutation length* $L_{\mathcal{P}}(F \vdash 0)$ and $\mathcal{P}$*-refutation space* $Sp_{\mathcal{P}}(F \vdash 0)$ of F are analogously defined by taking the minimum with respect to length or space, respectively, over all $\mathcal{P}$-refutations of F.

When the proof system in question is clear from context, we will drop the subindex in the proof complexity measures.

Let us now show how some proof systems that will be of interest to us can be defined in the framework of Definition 14.1. We remark that although we will not discuss this in any detail, all proof systems below are sound and implicationally complete, i.e., they can refute a CNF formula F if and only if F is unsatisfiable. Below, the notation $\frac{G_1 \quad \cdots \quad G_m}{H}$ means that if $G_1, \ldots, G_m$ have been derived previously in the proof (and are currently in memory), then we can infer H.

Definition 14.3 (k-DNF-resolution). The k*-DNF-resolution* proof systems are a family of sequential proof systems $\mathcal{R}(k)$ parameterized by $k \in \mathbb{N}^+$. Lines in a k-DNF-resolution refutation are k-DNF formulas and we have the following inference rules (where G, H denote k-DNF formulas, T, T' denote k-terms, and $a_1, \ldots, a_k$ denote literals):

***k*-cut** $\frac{(a_1 \wedge \cdots \wedge a_{k'}) \vee G \quad \overline{a}_1 \vee \cdots \vee \overline{a}_{k'} \vee H}{G \vee H}$, where $k' \leq k$.

$\wedge$-introduction $\frac{G \vee T \quad G \vee T'}{G \vee (T \wedge T')}$, as long as $|T \cup T'| \leq k$.

$\wedge$-elimination $\frac{G \vee T}{G \vee T'}$ for any $T' \subseteq T$.

Weakening $\frac{G}{G \vee H}$ for any k-DNF formula H.

For standard resolution, i.e., $\mathcal{R}(1)$, the k-cut rule simplifies to the *resolution rule*

$$\frac{B \vee x \quad C \vee \overline{x}}{B \vee C} \tag{14.3}$$

for clauses B and C. We refer to (14.3) 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. Clearly, in resolution the $\wedge$-introduction and $\wedge$-elimination rules do not apply. In fact, it can also be shown that the weakening rule never needs to be used in resolution refutations, but it is convenient to allow it in order to simplify some technical arguments in proofs.

For $\mathcal{R}(k)$-systems, the length measure is as defined in Definition 14.2, and for space we get the two measures *formula space* and *total space* depending on whether we consider the number of k-DNF formulas in a configuration or all literals in it, counted with repetitions. For standard resolution there are two more space-related measures that will be relevant, namely *width* and *variable space*. For clarity, let us give an explicit definition of all space-related measures for resolution that will be of interest.

Definition 14.4 (Width and space in resolution). The *width* $W(C)$ of a clause C is the number of literals in it, and the width of a CNF formula or clause configuration is the size of a widest clause in it. The *clause space* (as the formula space measure is known in resolution) $Sp(\mathbb{C})$ of a clause configuration $\mathbb{C}$ is $|\mathbb{C}|$, i.e., the number of clauses in $\mathbb{C}$, the *variable space*[7] $VarSp(\mathbb{C})$ is $|\mathit{Vars}(C)|$, i.e., the number of distinct variables mentioned in $\mathbb{C}$, and the *total space* $TotSp(\mathbb{C})$ is $\sum_{C \in \mathbb{C}} |C|$, i.e., the total number of literals in $\mathbb{C}$ counted with repetitions.

The width or space of a resolution refutation π is the maximum that the corresponding measures attains over any configuration $\mathbb{C} \in \pi$, and taking the minimum over all refutations of a formula F, we can define the width $W(F \vdash 0) = \min_{\pi : F \vdash 0}\{W(\pi)\}$, clause space $Sp(F \vdash 0) = \min_{\pi : F \vdash 0}\{Sp(\pi)\}$, variable space $VarSp(F \vdash 0) = \min_{\pi : F \vdash 0}\{VarSp(\pi)\}$, and total space $TotSp(F \vdash 0) = \min_{\pi : F \vdash 0}\{TotSp(\pi)\}$ of refuting F in resolution.

Restricting the refutations to tree-like resolution, we can define the measures $L_{\mathcal{T}}(F \vdash 0)$, $Sp_{\mathcal{T}}(F \vdash 0)$, $VarSp_{\mathcal{T}}(F \vdash 0)$, and $TotSp_{\mathcal{T}}(F \vdash 0)$ (note that width in general and tree-like resolution in the same, so defining tree-like width separately does not make much sense). However, in this paper we will almost exclusively focus on general, unrestricted versions of resolution and other proof systems.

Remark 14.5. When studying and comparing the complexity measures for resolution in Definition 14.4, as was noted in [ABRW02] it is preferable to prove the results for k-CNF formulas, i.e., formulas where all clauses have size upper-bounded by some constant. This is especially so since the width and space measures can "misbehave" rather artificially for formula families of unbounded width (see [Nor09b, Section 5] for a discussion of this point). Since every CNF formula can be rewritten as an equivalent formula of bounded width—in fact, as a 3-CNF formula, by using extension variables as described on page 128—it therefore seems natural to insist that the formulas under study should have width bounded by some constant.

[7] We remark that there is some terminological confusion in the literature here. The term "variable space" has also been used in previous papers (including by the current author) to refer to what is here called "total space." The terminology adopted in this paper is due to Alex Hertel and Alasdair Urquhart (see [Her08]), and we feel that although their naming convention is as of yet less well-established, it feels much more natural than the alternative.

Let us also give examples of some other propositional proof systems that have been studied in the literature, and that will be of some interest later in this survey. The first example is the *Cutting Planes* proof system, or *CP* for short, which was introduced in [CCT87] based on ideas in [Chv73, Gom63]. Here, clauses are translated to linear inequalities—for instance, $x \lor y \lor \overline{z}$ gets translated to $x + y + (1 - z) \geq 1$, i.e., $x + y - z \geq 0$—and these linear inequalities are then manipulated to derive a contradiction.

Definition 14.6 (Cutting Planes (CP)). Lines in a Cutting Planes proof are linear inequalities with integer coefficients. The derivation rules are as follows:

Variable axioms $\frac{}{x \geq 0}$ and $\frac{}{-x \geq -1}$ for all variables x.

Addition $\frac{\sum a_i x_i \geq A \quad \sum b_i x_i \geq B}{\sum (a_i + b_i) x_i \geq A + B}$

Multiplication $\frac{\sum a_i x_i \geq A}{\sum c a_i x_i \geq cA}$ for a positive integer c.

Division $\frac{\sum c a_i x_i \geq A}{\sum a_i x_i \geq \lceil A/c \rceil}$ for a positive integer c.

A CP-refutation ends when the inequality $0 \geq 1$ has been derived.

As shown in [CCT87], Cutting Planes is exponentially stronger than resolution with respect to length, since a CP-refutation can mimic any resolution refutation line by line and furthermore CP can easily handle the pigeonhole principle which is intractable for resolution. Exponential lower bounds on proof lengths for Cutting Planes were first proven in [BPR95] for the restricted subsystem CP* where all coefficients in the linear inequalities can be at most polynomial in the formula size, and were later extended to general CP in [Pud97]. To the best of our knowledge, it is open whether CP is in fact strictly stronger than CP* or not. We are not aware of any papers studying CP-space, but this was mentioned as an interesting open problem in [ABRW02].

The $\mathcal{R}(k)$-systems are *logic-based* proof systems in the sense that they manipulate logic formulas, and Cutting Planes is an example of a *geometry-based* proof systems where clauses are treated as geometric objects. Another class of proof systems is *algebraic* systems. One such proof system is *Polynomial Calculus (PC)*, which was introduced in [CEI96] under the name of "Gröbner proof system." Polynomial Calculus is interesting because there are provably efficient proof search algorithms for this system (where the performance of such algorithms for a formula F is measured in terms of the smallest possible proof there is to be found for F). In a PC-refutation, clauses are interpreted as multilinear polynomials. For instance, the requirement that the clause $x \lor y \lor \overline{z}$ should be satisfied gets translated to the equation $(1 - x)(1 - y)z = 0$ or $xyz - xz - yz + z = 0$, and we derive contradiction by showing that there is no common root for the polynomial equations corresponding to all the clauses.

Definition 14.7 (Polynomial Calculus (PC)). Lines in a Polynomial Calculus proof are multivariate polynomial equations $p = 0$, where $p \in \mathbb{F}[x, y, z, \ldots]$ for some (fixed) field $\mathbb{F}$. It is customary to omit "$= 0$" and only write p. The derivation rules are as follows, where $\alpha, \beta \in \mathbb{F}$, $p, q \in \mathbb{F}[x, y, z, \ldots]$, and x is any variable:

Variable axioms $\frac{}{x^2 - x}$ for all variables x (forcing 0/1-solutions).

Linear combination $\frac{p \quad q}{\alpha p + \beta q}$

Multiplication $\frac{p}{xp}$

A PC-refutation ends when 1 has been derived (i.e., $1 = 0$). The *size* of a PC-refutation is defined as the total number of monomials in the refutation. Another important measure is the *degree* of a refutation, which is the maximal (total) degree of any monomial.

The minimal refutation degree for a CNF formula F is closely related to the minimal refutation size. Impagliazzo et al. [IPS99] showed that every PC-proof of size S can be transformed into another PC-proof of degree $\mathrm{O}\big(\sqrt{n \log S}\big)$. A number of strong lower bounds on proof size have been obtained by proving degree lower bounds in, for instance, [AR01, BI99, BGIP01, IPS99, Raz98].

The representation of a clause $\bigvee_{i=1}^{n} x_i$ as a PC-polynomial is $\prod_{i=1}^{n}(1-x_i)$, which means that the number of monomials is exponential in the clause width. This makes Polynomial Calculus very weak with respect to space when compared to resolution, and arguably for a somewhat artificial reason. In [ABRW02], therefore, the proof system *Polynomial Calculus with Resolution (PCR)* was introduced as a common extention of Polynomial Calculus and resolution. The idea is to add an extra set of parallell formal variables $x', y', z', \ldots$ so that positive and negative literals can both be represented in a space-efficient fashion.

Definition 14.8 (Polynomial Calculus with Resolution (PCR)). Lines in a PCR-proof are polynomials over the ring $\mathbb{F}[x, x', y, y', z, z', \ldots]$, where as before $\mathbb{F}$ is some field. We have all the axioms and rules of PC plus the following axiom:

Complementarity $\frac{}{x+x'-1}$ for all pairs of variables (x, x').

Size and degree are defined as for Polynomial Calculus. The *monomial space* of a PCR-refutation is the number of *distinct* monomials in a configuration (i.e., not counted with repetitions).

The point of the complementarity rule is to force x and x' to have opposite values in $\{0, 1\}$, so that they encode complementary literals. One gets the same degree bounds for PCR as in PC (just substitute $1 - x$ for x'), but one can potentially avoid an exponential blow-up in size measured in the number of monomials (and thus also for space). Our running example clause $x \vee y \vee \overline{z}$ is rendered as $x'y'z$ in PCR.

The monomial space measure in PCR is meant to be compared to clause space in resolution. It was shown in [ABRW02] that PCR can save a constant factor in space with respect to resolution. We mention that another paper by the same authors studying PCR is [ABRW04].

It was observed in [ABRW02] the tight relation between degree and size in PC carries over to PCR. In a recent paper [GL10], Galesi and Lauria showed that this trade-off is essentially optimal, and also studied a generalization of PCR that unifies Polynomial Calculus and k-DNF resolution.

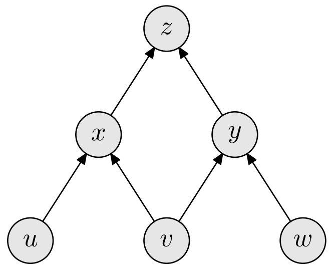
In general, the admissible inferences in a proof system according to Definition 14.1 is defined by a set of syntactic inference rules. In what follows, we will also be interested in a strengthened version of this concept, which was made explicit in [ABRW02].

Definition 14.9 (Syntactic and semantic derivations). We refer to derivations according to Definition 14.1, where each new line L has to be inferred by one of the inference rules for $\mathcal{P}$, as *syntactic* derivations. If instead *any line* L that is semantically implied by the current configuration can be derived in one atomic step, we talk about a *semantic*[8] derivation.

Clearly, semantic derivations are at least as strong as syntactic ones, and they are easily seen to be superpolynomially stronger with respect to length for any proof system where superpolynomial lower bounds are known. This is so since a semantic proof system can download all axioms in the formula one by one, and then deduce contradiction in one step since the formula is unsatisfiable. Therefore, semantic versions of proof systems are mainly interesting when we want to reason about space or about the relationship between space and length.

This concludes our presentation of proof systems, and we next turn to the connection between proof complexity and pebble games.

[8] It should be noted here that the term *semantic resolution* is usually used in the literature to refer to something very different, namely a restricted subsystem of (syntactic) resolution. In this section, however, semantic proofs will always be proofs in the sense of Definition 14.9.



(a) Pyramid graph Π_2 of height 2.

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

(b) Pebbling contradiction Peb_{Π_2}.

Figure 28: Pebbling contradiction for the pyramid graph Π_2.

14.2.3 Pebbling Contradictions and Substitution Formulas

The way pebbling results have been used in proof complexity has mainly been by studying so-called *pebbling contradictions* (also known as *pebbling formulas* or *pebbling tautologies*). These are CNF formulas encoding the pebble game played on a DAG G by postulating the sources to be true and the sink to be false, and specifying that truth propagates through the graph according to the pebbling rules. The idea to use such formulas seems to have appeared for the first time in Kozen [Koz77], and they were also studied in [RM99, BEGJ00] before being defined in full generality by Ben-Sasson and Wigderson in [BW01].

Definition 14.10 (Pebbling contradiction). Suppose that G is a DAG with sources S and a unique sink z. Identify every vertex $v \in V(G)$ with a propositional logic variable v. The *pebbling contradiction* over G, denoted Peb_G, is the conjunction of the following clauses:

- for all $s \in S$, a unit clause s (*source axioms*),
- For all non-source vertices v with immediate predecessors $pred(v)$, the clause $\bigvee_{u \in pred(v)} \overline{u} \vee v$ (*pebbling axioms*),
- for the sink z, the unit clause $\overline{z}$ (*target* or *sink axiom*).

If G has n vertices and maximal indegree ℓ, the formula Peb_G is a minimally unsatisfiable $(1+\ell)$-CNF formula with $n+1$ clauses over n variables. We will almost exclusively be interested in dags with bounded indegree $\ell = \mathrm{O}(1)$, usually $\ell = 2$. We note that DAGs with fan-in 2 and a single sink have sometimes been referred to as *circuits* in the proof complexity literature, although we will not use that term here. For an example of a pebbling contradiction, see the CNF formula in Figure 28(b) defined in terms of the graph in Figure 28(a).

In many of the cases we will be interested in below, the formulas in Definition 14.10 are not quite sufficient for our purposes since they are a bit too easy to refute. We therefore want to make them (moderately) harder, and it turns out that a good way of achieving this is to substitute some suitable Boolean function $f(x_1, \ldots, x_d)$ for each variable x and expand to get a new CNF formula.

It will be useful to formalize this concept of substitution for any CNF formula F and any Boolean function f. To this end, let f_d denote any (non-constant) Boolean function $f_d : \{0,1\}^d \mapsto \{0,1\}$ of arity d. We use the shorthand $\vec{x} = (x_1, \ldots, x_d)$, so that $f_d(\vec{x})$ is just an equivalent way of writing $f_d(x_1, \ldots, x_d)$. Every function $f_d(x_1, \ldots, x_d)$ is equivalent to a CNF formula over $x_1, \ldots, x_d$ with at most 2^d clauses. Fix some canonical set of clauses representing f_d and let $Cl[\neg f_d(\vec{x})]$ denote the clauses in some chosen canonical representation of the negation of f_d. This canonical representation can be given by a formal definition (in terms of min- and maxterms), but we do not want to get too formal here and instead try to convey the intuition by providing a few examples. For instance, we have

$$Cl[\vee_2(\vec{x})] = \{x_1 \vee x_2\} \quad \text{and} \quad Cl[\neg\vee_2(\vec{x})] = \{\overline{x}_1, \overline{x}_2\} \tag{14.4}$$

for logical or of two variables and

$$Cl[\oplus_2(\vec{x})] = \{x_1 \vee x_2, \overline{x}_1 \vee \overline{x}_2\} \quad \text{and} \quad Cl[\neg\oplus_2(\vec{x})] = \{x_1 \vee \overline{x}_2, \overline{x}_1 \vee x_2\} \tag{14.5}$$

for exclusive or of two variables. If we let thr^k_d denote the threshold function saying that k out of d variables are true, then for thr^2_4 we have

$$Cl[thr^2_4(\vec{x})] = \left\{\begin{matrix} x_1 \vee x_2 \vee x_3, \\ x_1 \vee x_2 \vee x_4, \\ x_1 \vee x_3 \vee x_4, \\ x_2 \vee x_3 \vee x_4 \end{matrix}\right\} \quad \text{and} \quad Cl[\neg thr^2_4(\vec{x})] = \left\{\begin{matrix} \overline{x}_1 \vee \overline{x}_2, \\ \overline{x}_1 \vee \overline{x}_3, \\ \overline{x}_1 \vee \overline{x}_4, \\ \overline{x}_2 \vee \overline{x}_3, \\ \overline{x}_2 \vee \overline{x}_4, \\ \overline{x}_3 \vee \overline{x}_4 \end{matrix}\right\} . \tag{14.6}$$

We want to define formally what it means to substitute f_d for the variables $Vars(F)$ in a CNF formula F. For notational convenience, we assume that F only has variables x, y, z, et cetera, without subscripts, so that $x_1, \ldots, x_d, y_1, \ldots, y_d, z_1, \ldots, z_d, \ldots$ are new variables not occurring in F.

Definition 14.11 (Substitution formula). For a positive literal x and a non-constant Boolean function f_d, define the *f_d-substitution* of x to be $x[f_d] = Cl[f_d(\vec{x})]$, i.e., the canonical representation of $f_d(x_1, \ldots, x_d)$ as a CNF formula. For a negative literal $\neg y$, the f_d-substitution is $\neg y[f_d] = Cl[\neg f_d(\vec{y})]$. The f_d-substitution of a clause $C = a_1 \vee \cdots \vee a_k$ is the CNF formula

$$C[f_d] = \bigwedge_{C_1 \in a_1[f_d]} \cdots \bigwedge_{C_k \in a_k[f_d]} (C_1 \vee \ldots \vee C_k) \tag{14.7}$$

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

For example, for the clause $C = x \vee \overline{y}$ and the exclusive or function $f_2 = x_1 \oplus x_2$ we have

$$\begin{aligned} C[f_2] = \; & (x_1 \vee x_2 \vee y_1 \vee \overline{y}_2) \wedge (x_1 \vee x_2 \vee \overline{y}_1 \vee y_2) \\ & \wedge (\overline{x}_1 \vee \overline{x}_2 \vee y_1 \vee \overline{y}_2) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{y}_1 \vee y_2) \ . \end{aligned} \tag{14.8}$$

Note that $F[f_d]$ is a CNF formula over $d \cdot |Vars(F)|$ variables containing strictly less than $|F| \cdot 2^{d \cdot W(F)}$ clauses. (Recall that we defined a CNF formula as a set of clauses, which means that $|F|$ is the number of clauses in F.) It is easy to verify that $F[f_d]$ is unsatisfiable if and only if F is unsatisfiable.

Two examples of substituted version of the pebbling formula in Figure 28(b) are the substitution with logical or in Figure 29(a) and with exclusive or in Figure 29(b). As we shall see, these formulas have played an important role in the line of research trying to understand proof space in resolution. For our present purposes, there is an important difference between logical or and exclusive or which is captured by the next definition.

Definition 14.12 (Non-authoritarian function [BN11]). We say that a Boolean function $f(x_1, \ldots, x_d)$ is *k-non-authoritarian* if no restriction to $\{x_1, \ldots, x_d\}$ of size k can fix the value of f. In other words, for every restriction ρ to $\{x_1, \ldots, x_d\}$ with $|\rho| \leq k$ there exist two assignments $\alpha_0, \alpha_1 \supset \rho$ such that $f(\alpha_0) = 0$ and $f(\alpha_1) = 1$. If this does not hold, f is *k-authoritarian*. A 1-(non-)authoritarian function is called just *(non-)authoritarian*.

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

(a) Substitution pebbling contradiction $Peb_{\Pi_2}[\lor_2]$ with respect to binary logical or.

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

(b) Substitution pebbling contradiction $Peb_{\Pi_2}[\oplus_2]$ with respect to binary exclusive or.

Figure 29: Examples of substitution pebbling formulas for the pyramid graph Π_2.

Observe that a function on d variables can be k-non-authoritarian only if $k < d$. Two natural examples of d-non-authoritarian functions are exclusive or $\oplus$ of $d+1$ variables and majority of $2d+1$ variables, i.e., thr^{d+1}_{2d+1}. Non-exclusive or of any arity is easily seen to be an authoritarian function, however, since setting any variable x_i to true forces the whole disjunction to true.

Concluding our presentation of preliminaries, we remark that the idea of combining Definition 14.10 with Definition 14.11 was not a dramatic new insight originating with [BN08], but rather the natural generalization of ideas in many previous articles. For instance, the papers [BIW04, Ben09, BIPS10, BW01, BP07, ET03, Nor09a, NH08b] all study formulas $Peb_G[\vee_2]$, and [EGM04] considers formulas $Peb_G[\wedge_l \vee_k]$. And in fact, already back in 2006 Atserias [Ats06] proposed that XOR-pebbling contradictions $Peb_G[\oplus_2]$ could potentially be used to separate length and space in resolution, as was later shown to be the case in [BN08].

14.3 Overview of Pebbling Contradictions in Proof Complexity

Let us now give a general overview of how pebbling contradictions have been used in proof complexity. While we have striven to give a reasonably full picture below, we should add the caveat that our main focus is on resolution-based proof systems, i.e., standard resolution and $\mathcal{R}(k)$ for $k > 1$. Also, to avoid confusion it should be pointed out (again) that the pebble games examined here should not be mixed up with the very different *existential pebble games* which have also proven to be a useful tool in proof complexity in, for instance, [Ats04, AKV04, BG03, GT05] and in this context perhaps most notably in the paper [AD08] establishing the upper bound $Sp(F \vdash 0) \geq W(F \vdash 0) - \mathrm{O}(1)$ on width in terms of clause space for k-CNF formulas F.

We have divided the overview into four parts covering (a) questions about time-space trade-offs and separations, (b) comparisons of proof systems and subsystems of proof systems, (c) formulas used as benchmarks for SAT solvers, and (d) the computational complexity of various proof measures. In what follows, our goal is to survey the results in fairly non-technical terms. A more detailed discussion of the techniques used to prove results on time and space will follow in Section 14.4.

14.3.1 Time Versus Space

As we have seen in this survey, pebble games have been used extensively as a tool to prove time and space lower bounds and trade-offs for computation. One can ask if it could be the case that when we encode pebble games in terms of CNF formulas, these formulas should inherit the same properties as the underlying graphs. That is, if we start with a DAG G such that any pebbling of G in short time must have large pebbling space, can we argue that the corresponding pebbling contradiction should have the property that any short resolution refutation of this formula must also require large proof space?

In one direction the correspondence between pebbling and resolution is straightforward. As was observed in [BIW04], if there is a black pebbling strategy for the graph G in time τ and space s, then the CNF formula Peb_G can be refuted by resolution in length $\mathrm{O}(\tau)$ and space $\mathrm{O}(s)$.

The other direction is much less obvious. 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_G we should be able to extract a pebbling of the DAG G. To formalize this intuition, we would like to prove something 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_G in terms of 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_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 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 space lower bounds and time-space trade-offs for pebble games, we would then be able to lift such bounds and trade-offs to resolution.

The first important step towards realizing the above program was taken by Ben-Sasson in 2002 (journal version in [Ben09]), who was the first to prove trade-offs between proof complexity measures in resolution. The key insight in [Ben09] was to interpret resolution refutations of Peb_G in terms of black-white pebblings of G, by letting positive literals on the blackboard correspond to black pebbles and negative literals to white pebbles. One can then show that using this correspondence (and modulo some technicalities), any resolution refutation of Peb_G results in a black-white pebbling of G in pebbling time upper-bounded by the refutation length and pebbling space upper-bounded by the refutation *variable* space (Definition 14.4).

This translation of refutations to black-white pebblings was used by Ben-Sasson to establish strong trade-offs between clause space and width in resolution. He showed that there are k-CNF formulas F_n of size $\Theta(n)$ which can be refuted both in constant clause space $Sp(F_n \vdash 0)$ and in constant width $W(F_n \vdash 0)$, but for which any refutation π_n that tries to optimize both measures simultaneously can never do better than $Sp(\pi_n) \cdot W(\pi_n) = \Omega(n/\log n)$. This result was obtained by studying formulas Peb_G over the graphs G in [GT78] with black-white pebbling price $\textsf{\textit{BW-Peb}}(G) = \Omega(n/\log n)$. Since the upper bounds $Sp(\pi) \cdot W(\pi) \geq TotSp(\pi) \geq VarSp(\pi)$ are easily seen to hold for any resolution refutation π, and since by what was just said we must have $VarSp(\pi_n \vdash 0) = \Omega(n/\log n)$, one gets the space-width trade-off stated above. In a separate argument, one shows that $Sp\big(Peb_G \vdash 0\big) = \mathrm{O}(1)$ and $W\big(Peb_G \vdash 0\big) = \mathrm{O}(1)$. Using the same ideas plus upper bound on space in terms of size in [ET01], [Ben09] also proved that for tree-like resolution it holds that $L_{\mathcal{T}}\big(Peb_G \vdash 0\big) = \mathrm{O}(n)$ and $W\big(Peb_G \vdash 0\big) = \mathrm{O}(1)$ but for any particular tree-like refutation π_n there is a length-width trade-off $W(\pi_n) \cdot \log L(\pi_n) = \Omega(n/\log n)$.

Unfortunately, the results in [Ben09] also show that the program outlined above for proving time-space trade-offs will *not* work for general resolution. This is so since for any DAG G the formula Peb_G is refutable in linear length and constant clause space simultaneously. It turns out that what we can do instead is to look at substitution formulas $Peb_G[f]$ for suitable Boolean functions f, but this leads to a number of technical complications. However, building on previous works [Nor09a, NH08b], a way was finally found to realize the program in [BN11]. We will give a more detailed exposition of the proof techniques in Section 14.4, but let us conclude this discussion of time-space trade-offs by describing the flavour of the results obtained in these latter papers.

Let $\{G_n\}_{n=1}^{\infty}$ be a family of single-sink DAGs of size $\Theta(n)$ and with bounded fan-in. Suppose that there are functions $s_{\mathrm{lo}}(n) \ll s_{\mathrm{hi}}(n) = \mathrm{O}(n/\log\log n)$ such that G_n has black pebbling price $\textsf{\textit{Peb}}(G_n) = s_{\mathrm{lo}}(n)$ and there are black-only pebbling strategies for G_n in time $\mathrm{O}(n)$ and space $s_{\mathrm{hi}}(n)$, but any black-white pebbling strategy in space $\mathrm{o}(s_{\mathrm{hi}}(n))$ must have superpolynomial or even exponential length. Also, let K be a fixed positive integer. Then there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\mathrm{O}(n)$ and width $\mathrm{O}(1)$ (with constants depending on K) such that the following holds:

- The formulas F_n are refutable in syntactic resolution in (small) total space $\mathrm{O}(s_{\mathrm{lo}}(n))$.
- There are also syntactic resolution refutations π_n of F_n in simultaneous length $\mathrm{O}(n)$ and (much larger) total space $\mathrm{O}(s_{\mathrm{hi}}(n))$.
- However, any resolution refutation, even semantic, in formula space $\mathrm{o}(s_{\mathrm{hi}}(n))$ must have superpolynomial or sometimes even exponential length.

- Even for the much stronger semantic k-DNF resolution proof systems, $k \leq K$, it holds that any $\mathcal{R}$(k)-refutation of F_n in formula space $\mathrm{o}\big(\sqrt[k+1]{s_{\mathrm{hi}}(n)}\big)$ must have superpolynomial length (or exponential length, correspondingly).

This "theorem template" can be instantiated for a wide range of space functions $s_{\mathrm{lo}}(n)$ and $s_{\mathrm{hi}}(n)$, from constant space all the way up to nearly linear space, using graph families with suitable trade-off properties (for instance, those in Sections 8, 9, 10, and 11). Also, absolute lower bounds on black-white pebbling space, such as in Section 7, yield corresponding lower bounds on clause space.

Moreover, these trade-offs are robust in that they are not sensitive to small variations in either length or space. The way we would like to think about this, with some handwaving intuition, is that the trade-offs will not show up only for a SAT solver being unlucky and picking just the wrong threshold when trying to hold down the memory consumption. Instead, any resolution refutation having length or space in the same general vicinity will be subject to the same qualitative trade-off behaviour.

14.3.2 Separations of Proof Systems

A number of restricted subsystems of resolution, often referred to as *resolution refinements*, have been studied in the proof complexity literature. These refinements were introduced to model SAT solvers that try to make the proof search more efficient by narrowing the search space, and they are defined in terms of restrictions on the DAG representations G_π of resolution refutations π. An interesting question is how the strength of these refinements are related to one another and to that of general, unrestricted resolution, and pebbling has been used as a tool in several papers investigating this. We briefly discuss some of these restricted subsystems below, noting that they are all known to be sound and complete. We remark that more recently, a number of different (but related) models for the proof system underlying state-of-the-art DPPL SAT solvers with *clause learning* have also been proposed and studied theoretically in [BKS04, BJ10, BHJ08, HBPV08, Van05] but going into details here is unfortunately outside the scope of this survey.

A *regular resolution* refutation of a CNF formula F is a refutation π such that on any path in G_π from an axiom clause in F to the empty clause 0, no variable is resolved over more than once. We call a regular resolution refutation *ordered* if in addition there exists an ordering of the variables such that every sequence of variables labelling a path from an axiom to the empty clause respects this ordering. Ordered resolution is also known as *Davis-Putnam resolution*. A *linear resolution* refutation is a refutation π with the additional restriction that the underlying DAG G_π must be linear. That is, the proof should consist of a sequence of clauses $\{C_1, C_2, \ldots, C_m = 0\}$ such that for every $i \in [m]$ it holds for the clause C_i that it is either an axiom clause of F or is derived from C_{i-1} and C_j for some $j < i$ (where C_j can be an axiom clause). Finally, as was already mentioned in Definition 14.1, a *tree-like* refutation is one in which the underlying DAG is a tree. Tree-like resolution is also called *Davis-Logemann-Loveland* or *DLL resolution* in the literature. The reason for this is that tree-like resolution refutations can be shown to correspond to refutations produced by the proof search algorithm in [DLL62], known as DLL or DPLL, that fixes one variable x in the formula F to true or false respectively, and then redursively tries to refute the two formulas corresponding to the two values of x (after simplifications, i.e., removing satisfied clauses and shrinking clauses with falsified literals).

It is known that tree-like resolution proofs can always be assumed to be regular without any loss of generality [Urq95], and clearly ordered refutations are regular by definition. Alekhnovich et al. [AJPU07] established an exponential separation with respect to length between general and regular resolution, and Bonet et al. [BEGJ00] showed that tree-like resolution can be exponentially weaker than ordered resolution and some other resolution refinements. Johannsen [Joh01] proved that tree-like and ordered resolution can be exponentially separated, from which it follows that regular and ordered resolution can be exponentially separated as well and that tree-like and ordered resolution are incomparable. More separations for other resolution refinements not mentioned above were presented in [BP07], but an exposition of these results are outside the scope of this survey.

The construction in [AJPU07] uses an implicit encoding of the pebbling formulas in Definition 14.10 in the sense that they study formulas encoding that each vertex in the DAG contains a pebble, identified by a unique number. For every pebble, there is a variable encoding the colour of this pebble—red or blue—where source vertices are known to have red pebbles and the sink vertex should have a blue one. Finally, there are clauses enforcing that if all predecessors of a vertex has red pebbles, then the pebble on that vertex must be red. These formulas can be refuted bottom-up in linear length just as our standard pebbling contradictions, but such refutations are highly irregular. The paper [BEGJ00], which also presents lower bounds for tree-like CP-proofs for formulas easy for resolution, uses another variant of pebbling contradictions defined over pyramid graphs, but we omit the details. Later, [BIW04] proved a stronger exponential separation of general and tree-like resolution, improving on the separation implied by [BEGJ00], and this latter paper uses substitution pebbling contradictions $Peb_G[\vee_2]$ and the $\Omega(n/\log n)$ lower bound on black pebbling in [PTC77] (see Section 7).

Intriguingly, linear resolution is *not* known to be weaker then general resolution. The conventional wisdom seems to be that linear resolution should indeed be weaker, but the difficulty is that if so it can only be weaker on a technicality. Namely, it was shown in [BP07] that if a polynomial number of appropriately chosen tautological clauses are added to any CNF formula, then linear resolution can simulate general resolution by using these extra clauses. Any separation would therefore have to argue very "syntactically."

Esteban et al. [EGM04] showed that tree-like k-DNF resolution proof systems form a strict hierarchy with respect to proof length and proof space. The space separation they obtain is for formulas requiring formula space $\mathrm{O}(1)$ in $\mathcal{R}(k+1)$ but formula space $\mathrm{O}(n/\log^2 n)$ in $\mathcal{R}(k)$. Both of these separation results use a special flavour $Peb_G[\wedge_l \vee_k]$ of substitution pebbling formulas, again defined over the graphs G in [PTC77] with black pebbling price $\Omega(n/\log n)$. As was mentioned above, the space separation was strengthened to general, unrestricted $\mathcal{R}(k)$-systems in [BN11], but with worse parameters. This result uses formulas $Peb_G[\oplus_{k+1}]$ defined in terms of exclusive or of $k+1$ variables to get the separation between $\mathcal{R}(k+1)$ and $\mathcal{R}(k)$, as well as the stronger lower bound $\Omega(n/\log n)$ for *black-white* pebbling in [GT78].

Concluding our discussion of separation of resolution refinements, we also want to mention that Esteban and Torán [ET03] used substitution pebbling contradictions $Peb_G[\vee_2]$ over complete binary trees to prove that general resolution is strictly stronger than tree-like resolution with respect to clause space. Expressed in terms of formula size the separation one obtains is in the constant multiplicative factor in front of the logarithmic space bound. This might not sound too impressive, but recall that the space complexity it at most linear in the number of variables and clauses, so it makes sense to care about constant factors here. Also, it should be noted that this paper had quite some impact in that the technique used to establish the separation can be interpreted as a (limited) way of of simulating black-white pebbling in resolution, and this provided one of the key insights for [Nor09a] and the ensuing papers considered in Section 14.3.1.

14.3.3 Benchmark Formulas

Pebbling contradictions have also been used as benchmark formulas for evaluating and comparing different proof search heuristics. Ben-Sasson et al. [BIW04] used the exponential lower bound discussed above for tree-like resolution refutations of formulas $Peb_G[\vee_2]$ to show that a proof search heuristic that exhaustively searches for resolution refutations in minimum width can sometimes be exponentially faster than DLL-algorithms searching for tree-like resolutions, while it can never be too much slower. Sabharwal et al. [SBK04] also used pebbling formulas to evaluate heuristics for clause learning algorithms. In a more theoretical work, Beame et al. [BIPS10] again used pebbling formulas $Peb_G[\vee_2]$ to compare and separate extensions of the resolution proof system using "formula caching," which is a generalization of clause learning.

In view of the strong length-space trade-offs for resolution, which were hinted at in Section 14.3.1 and will be examined in more detail below, a very natural question is whether these theoretical results also translate into trade-offs between time and space in practice for state-of-the-art SAT solvers using clause learning. Although the model in Definitions 14.1 and 14.2 for measuring time and space of resolution proofs is very simple, it still does not seem too unreasonable that it should be able to capture the problem in clause learning of which of the learned clauses should be kept in the clause database (which would roughly correspond to configurations in our refutations). It would be very interesting to take graphs G as in Sections 8 and 9, or possibly as in Sections 10 and 11 although these constructions are more complex and therefore perhaps not as good candidates, and study formulas $Peb_G[\vee_2]$ or $Peb_G[\oplus_2]$ over these graphs. For binary exclusive or $\oplus_2$ we have provable length-space trade-offs in terms of pebbling trade-offs for the corresponding DAGs, and although we cannot prove it, we strongly suspect that the same should hold true also for formulas defined in terms of usual logical or.

Open Problem 10. *Do pebbling contradictions $Peb_G[\vee_2]$ or $Peb_G[\oplus_2]$ exhibit time-space trade-offs for SAT solvers with clause learning similar to the pebbling trade-offs of the underlying DAGs G?*

Let us try to present a very informal argument why the answer to this question could be positive. On the one hand, all the length-space trade-offs that have been established so far are for sublinear space, and given that linear space is needed just to keep the formula in memory such space bounds might not seem to relevant for real-life applications. On the other hand, 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, in one sense one can argue that pebbling contradictions have already been shown to 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 one way of interpreting the experimental results in [SBK04], is that one of the state-of-the-art SAT solvers at that time had serious problems with even moderately large pebbling contradictions. Namely, the "grid pebbling formulas" in [SBK04] are precisely our OR-pebbling contradictions $Peb_G[\vee_2]$ over pyramids. Although we are certainly not arguing that this is the whole story—it was also shown in [SBK04] 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.

14.3.4 Complexity of Decision Problems

A number of papers have also used pebble games to study how hard it is to decide the complexity of a CNF formula F with respect to some proof complexity measure M. This is formalized in terms of decision problems as follows: "Given a CNF formula F and a parameter p, is there a refutation π of F with $M(\pi) \leq p$?"

The one proof complexity measure that is reasonably well understood is proof length. It has been shown (using techniques not related to pebbling) that the problem of finding a shortest refutation of a CNF formulas is NP-hard [Iwa97] and remains hard even if we just want to approximate the minimum refutation length [ABMP01].

With regard to proof space, Alex Hertel and Alasdair Urquhart [HU07] showed that tree-like resolution clause space is PSPACE-complete, using the exact combinatorial characterization of tree-like resolution clause space given in [ET03] and the generalized pebble game in [Lin78] mentioned in Section 13.2. They also proved (see [Her08, Chapter 6]) that variable space in general resolution is PSPACE-hard, although this result requires CNF formulas of unbounded width. Interestingly, variable space is *not* known to be in PSPACE, and the best upper bound obtained in [Her08] is that the problem is at least contained in EXPSPACE.

Another very interesting space-related result is that of Philipp Hertel and Toni Pitassi [HP07], who presented a PSPACE-completeness result for total space in resolution as well as some sharp trade-offs (in the sense of Section 13.1) for length with respect to total space, building on their PSPACE-completeness result for black-white pebbling mentioned in Section 13.2 and using the original pebbling contradictions Peb_G in Definition 14.10. Their construction is highly nontrivial, and unfortunately a bug was later found in the proofs leading to these results being withdrawn in the journal version [HP10] of the paper. Their trade-off results were later subsumed by those in [Nor09b], using other techniques not related to pebbling, but it remains open whether total space is PSPACE-complete or not (the problem is fairly easily seen to be in PSPACE).

Open Problem 11. *Given a CNF formula F (preferably of fixed width) and a parameter s, is it* PSPACE-*complete to determine whether F can be refuted in resolution in total space at most s?*

There are a number of other interesting open questions regarding the hardness of proof complexity measures for resolution. An obvious question is whether the PSPACE-completeness result for tree-like resolution clause space in [HU07] can be extended to clause space in general resolution. (Again, showing that clause space is PSPACE is relatively straightforward.)

Open Problem 12. *Given a CNF formula F (preferably of fixed width) and a parameter s, is it* PSPACE-*complete to determine whether F can be refuted in resolution in clause space at most s?*

A somewhat related question is whether it is possible to find a clean, purely combinatorial characterization of clause space. This has been done for resolution width [AD08] and tree-like resolution clause space [ET03], and this latter result was a key component in proving the PSPACE-completeness of tree-like space. It would be very interesting to find similar characterizations of clause space in general resolution and $\mathcal{R}(k)$.

Open Problem 13 ([ET03, EGM04]). *Is there a combinatorial characterization of clause space in general, unrestricted resolution? In k-DNF resolution?*

The complexity of determining resolution width is also open.

Open Problem 14. *Given a k-CNF formula F and a parameter w, is it* EXPTIME-*complete to determine whether F can be refuted in resolution in width at most w?*

The width measure was conjectured to be EXPTIME-complete by Moshe Vardi. As shown in [HU06], using the combinatorial characterization of width in [AD08], width is in EXPTIME. The paper [HU06] also claimed an EXPTIME-completeness result, but this has later been retracted [HU09]. The conclusion that can be drawn from all of this is perhaps that space is indeed a very tricky concept in proof complexity, and that we do not really understand space-related measures very well, even for such a simple proof system as resolution.

14.4 Translating Time-Space Trade-offs from Pebbling to Resolution

So far, we have discussed in fairly non-technical terms how pebble games have been used to prove different results in proof complexity. In this section, we want to elaborate on the length-space trade-off results for resolution-based proof systems mentioned in Section 14.3.2 and try to give a taste of how they are proven. Recall that the general idea is to establish reductions between pebbling strategies for DAGs on the one hand and refutations of corresponding pebbling contradictions on the other. We start by describing the reductions from pebblings to refutations in Section 14.4.1, and then examine how refutations can be translated to pebblings in Section 14.4.2.

14.4.1 Techniques for Upper Bounds on Resolution Trade-offs

Given any black-only pebbling $\mathcal{P}$ of a DAG G with bounded fan-in ℓ, it is straightforward to simulate this pebbling in resolution to refute the corresponding pebbling contradiction $Peb_G[f_d]$ in length $\mathrm{O}\big(\mathit{time}(\mathcal{P})\big)$ and space $\mathrm{O}\big(\mathit{space}(\mathcal{P})\big)$. This was perhaps first noted in [BIW04] for the simple Peb_G formulas, but the simulation extends readily to any formula $Peb_G[f_d]$, with the constants hidden in the asymptotic notation depending only on f_d and ℓ. In view of the translation in [Ben09] and subsequent works of resolution refutations to *black-white* pebblings, it is natural to ask whether this reduction goes both ways, i.e., whether resolution can simulate not only black pebblings but also black-white ones.

At first sight, it seems that resolution would have a hard time simulating black-white pebbling. To see why, let us start by considering a black-only pebbling $\mathcal{P}$. We can easily mimic such a pebbling in a resolution refutation of $Peb_G[f_d]$ by deriving that $f_d(v_1, \ldots, v_d)$ is true whenever the corresponding vertex v in G is black-pebbled. If the pebbling strategy places a pebble on v at time t, then we know that all predecessors of v have pebbles at this point. By induction, this implies that for all $w \in \mathit{pred}(v)$ we have clauses $w[f_d]$ in the configuration $\mathbb{C}_t$ encoding that all $f_d(w_1, \ldots, w_d)$ are true, and if we download the pebbling axioms for v we can derive the clauses $v[f_d]$ encoding that $f_d(v_1, \ldots, v_d)$ is true by the implicational completeness of resolution. Furthermore, this derivation can be carried out in length and extra clause space $\mathrm{O}(1)$, where the hidden constants depend only on ℓ and f_d as stated above. We end up deriving that $f_d(z_1, \ldots, z_d)$ is true for the sink z, at which point we can download the sink axioms and derive a contradiction.

The intuition behind this translation is that a black pebble on v means that we know v, which in resolution translates into truth of v. In the pebble game, having a white pebble on v instead means that we need to assume v. By duality, it seems reasonable to let this correspond to falsity of v in resolution. Focusing on the pyramid Π_2 in Figure 28(a), and pebbling contradiction $Peb_{\Pi_2}[\vee_2]$ in Figure 29(a), our intuitive understanding then becomes that white pebbles on x and y and a black pebble on z should correspond to the set of clauses

$$\{\overline{x}_i \vee \overline{y}_j \vee z_1 \vee z_2 \mid i,j = 1,2\} \tag{14.9}$$

which indeed encode that assuming $x_1 \vee x_2$ and $y_1 \vee y_2$, we can deduce $z_1 \vee z_2$. See Figure 30(a) for an illustration of this.

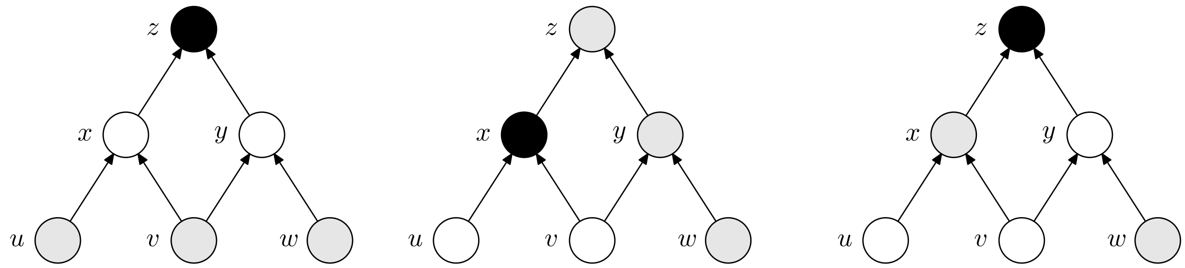
If we now place white pebbles on u and v, this allows us to remove the white pebble from x. Rephrasing this in terms of resolution, we can say that x follows if we assume u and v, which is encoded as the set of clauses

$$\{\overline{u}_i \vee \overline{v}_j \vee x_1 \vee x_2 \mid i,j = 1,2\} \tag{14.10}$$

in Figure 30(b), and indeed, from the clauses in (14.9) and (14.10) we can derive in resolution that z is black-pebbled and u, v and y are white pebbled, i.e., the set of clauses

$$\{\overline{u}_i \vee \overline{v}_j \vee \overline{y}_k \vee z_1 \vee z_2 \mid i,j,k = 1,2\} \tag{14.11}$$

in Figure 30(c). The above toy example indicates one of the problems one runs into when one tries to simulate black-white pebbling in resolution: as the number of white pebbles grows, there is an exponential



(a) $\{\overline{x}_i \vee \overline{y}_j \vee z_1 \vee z_2 \mid i,j = 1,2\}$. (b) $\{\overline{u}_i \vee \overline{v}_j \vee x_1 \vee x_2 \mid i,j = 1,2\}$. (c) $\{\overline{u}_i \vee \overline{v}_j \vee \overline{y}_k \vee z_1 \vee z_2 \mid i,j,k = 1,2\}$.

Figure 30: Black and white pebbles and (intuitively) corresponding sets of clauses.

blow-up in the number of clauses. The clause set in (14.11) is twice the size of those in (14.9) and (14.10), although it corresponds to only one more white pebble. What this suggests is that as a pebbling starts to make heavy use of white pebbles, resolution will not be able to mimic such a pebbling in a length- and space-preserving manner. This leads to the thought that perhaps black pebbling provides not only upper but also lower bounds on resolution refutations of pebbling contradictions.

However, it was recently shown in [Nor10a] that at least in certain instances, resolution can in fact be strictly better than black-only pebbling, both for time-space trade-offs and with respect to space in absolute terms. What is done in [Nor10a] is to design a limited version of black-white pebbling, tailor-made for resolution, where one explicitly restricts the amount of nondeterminism, i.e., white pebbles, a pebbling strategy can use. Such restricted pebbling use "few white pebbles per black pebble" (in a sense that will be made formal below), and can therefore be simulated in a time- and space-preserving fashion by resolution, avoiding the exponential blow-up just discussed. This game is essentially just a formalization of the naive simulation sketched above, but before stating the formal definitions, let us try to provide some intuition why the rules of this new game look the way they do.

First, if we want a game that can be mimicked by resolution, then placements of isolated white vertices do not make much sense. What a resolution derivation can do is to download axiom clauses, and intuitively this corresponds to placing a black pebble on a vertex together with white pebbles on its immediate predecessors, if it has any. Therefore, we adopt such aggregate moves as the only admissible way of placing new pebbles. For instance, looking at Figure 30 again, placing a black pebble on z and white pebbles on x and y corresponds to downloading the axiom clauses in (14.9) for $Peb_{\Pi_2}[\vee_2]$.

Second, note that if we have a black pebble on z with white pebbles on x and y corresponding to the clauses in (14.9) and a black pebble on x with white pebbles on u and v corresponding to the clauses in (14.10), we can derive the clauses in (14.11) corresponding to z black-pebbled and u, v and y white-pebbled but no pebble on x. This suggests that a natural rule for white pebble removal is that a white pebble can be removed from a vertex if a black pebble is placed *on that same vertex* (and not on its immediate predecessors).

Third, if we then just erase all clauses in (14.11), this corresponds to all pebbles disappearing. On the face of it, this is very much unlike the rule for white pebble removal in the standard pebble game, where it is absolutely crucial that a white pebble can only be removed when its predecessors are pebbled. However, the important point here is that not only do the white pebbles disappear—the black pebble that has been placed on z with the help of these white pebbles disappears as well. What this means is that we cannot treat black and white pebbles in isolation, but we have to keep track of for each black pebble which white pebbles it depends on, and make sure that the black pebble also is erased if any of the white pebbles supporting it is erased. The way we do this is to label each black pebble v with its supporting white pebbles W, and define the pebble game in terms of moves of such labelled *pebble subconfigurations* $v\langle W\rangle$.

Formalizing the loose description above, our pebble game is then defined as follows.

Definition 14.13 (Labelled pebbling [Nor10a]). For v a vertex and W a set of vertices in a DAG G, we say that $v\langle W\rangle$ is a *pebble subconfiguration* with a black pebble on v supported by white pebbles on all $w \in W$. The black pebble on v in $v\langle W\rangle$ is said to be *dependent* on the white pebbles in its *support* W. We refer to $v\langle\emptyset\rangle$ as an *independent black pebble*.

For G any DAG with unique sink z, a (complete) *labelled pebbling* of G is a sequence $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ of labelled pebble configurations such that $\mathbb{L}_0 = \emptyset$, $\mathbb{L}_\tau = \{z\langle\emptyset\rangle\}$, and for all $t \in [\tau]$ it holds that $\mathbb{L}_t$ can be obtained from $\mathbb{L}_{t-1}$ by one of the following rules:

Introduction $\mathbb{L}_t = \mathbb{L}_{t-1} \cup \{v\langle \mathit{pred}(v)\rangle\}$, where $\mathit{pred}(v)$ is the set of immediate predecessors of v.

Merger $\mathbb{L}_t = \mathbb{L}_{t-1} \cup \big\{v\langle (V \cup W) \setminus \{w\}\rangle\big\}$ for $v\langle V\rangle, w\langle W\rangle \in \mathbb{L}_{t-1}$ with $w \in V$. We denote this subconfiguration $\mathsf{merge}(v\langle V\rangle, w\langle W\rangle)$, and refer to it as a *merger on* w.

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

Let $Bl(\mathbb{L}_t) = \bigcup\{v \mid v\langle W\rangle \in \mathbb{L}_t\}$ denote the set of all black-pebbled vertices in $\mathbb{L}_t$ and $Wh(\mathbb{L}_t) = \bigcup\{W \mid v\langle W\rangle \in \mathbb{L}_t\}$ the set of all white-pebbled vertices. Then the space of an labelled pebbling $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ is $\max_{\mathbb{L}\in\mathcal{L}}\{|Bl(\mathbb{L}) \cup Wh(\mathbb{L})|\}$ and the time of $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ is $\mathsf{time}(\mathcal{L}) = \tau$.

The game in Definition 14.13 might look quite different from the standard black-white pebble game, but it is not hard to show that labelled pebblings are essentially just a restricted form of black-white pebblings. (We refer to [Nor10a] for formal proofs of this and all following claims in Section 14.4.1).

Lemma 14.14 ([Nor10a]). *If G is a single-sink DAG and $\mathcal{L}$ is a complete labelled pebbling of G, then there is a complete black-white pebbling $\mathcal{P}_\mathcal{L}$ of G with $\mathsf{time}(\mathcal{P}_\mathcal{L}) \leq \frac{4}{3}\mathsf{time}(\mathcal{L})$ and $\mathsf{space}(\mathcal{P}_\mathcal{L}) \leq \mathsf{space}(\mathcal{L})$.*

However, the definition of space of labelled pebblings does not seem quite right from the point of view of resolution. Not only does the space measure fail to capture the the exponential blow-up in the number of white pebbles discussed above. We also have the problem that if one white pebble is used to support many different black pebbles, then in a resolution refutation simulating such a pebbling we have to pay multiple times for this single white pebble, once for every black pebble supported by it. To get something that can be simulated by resolution, we therefore need to restrict the labelled pebble game even further.

Definition 14.15 (Bounded labelled pebblings [Nor10a]). An (m, S)*-bounded labelled pebbling* is a labelled pebbling $\mathcal{L} = \{\mathbb{L}_0, \ldots, \mathbb{L}_\tau\}$ such that every $\mathbb{L}_t$ contains at most m pebble subconfigurations $v\langle W\rangle$ and every $v\langle W\rangle$ has white support size $|W| \leq S$.

Observe that if a graph G with fan-in ℓ has a black-only pebbling strategy in time τ and space s, then the labelled pebbling simulating this strategy is an $(s, \ell + 1)$-bounded pebbling in time at most $\tau(\ell + 1)$. Thus, the power of bounded labelled pebbling is somewhere in between black-only and black-white pebbling.

Note also that boundedness automatically implies low space complexity, since an (m, S)-bounded labelled pebbling $\mathcal{L}$ clearly satisfies $\mathsf{space}(\mathcal{L}) \leq m(S + 1)$. And if we can design an (m, S)-bounded pebbling for a graph G, then such a pebbling can be used to refute any pebbling contradiction $Peb_G[f]$ in resolution by mimicking $\mathcal{L}$.

Lemma 14.16 ([Nor10a]). *Suppose that $\mathcal{L}$ is any complete (m, S)-bounded pebbling of a DAG G and that $f : \{0,1\}^d \mapsto \{0,1\}$ is any nonconstant Boolean function. Then there is a resolution refutation $\pi_\mathcal{L}$ of $Peb_G[f]$ in length $L(\pi_\mathcal{L}) = \mathsf{time}(\mathcal{L}) \cdot \exp\big(\mathrm{O}(dS)\big)$ and total space $TotSp(\pi_\mathcal{L}) = m \cdot \exp\big(\mathrm{O}(dS)\big)$. In particular, fixing f it holds that resolution can simulate $(m, \mathrm{O}(1))$-bounded pebblings in a time- and space-preserving manner.*

The whole problem thus boils down to the question whether there are graphs with (a) asymptotically different properties for black and black-white pebbling for which (b) optimal black-white pebblings can be carried out in the bounded labelled pebbling framework. The answer to this question is positive, and using Lemma 14.16 one can prove that resolution can be strictly better than black-only pebbling, both for time-space trade-offs and with respect to space in absolute terms. It turns out that for all known separation results in the pebbling literature where black-white pebbling does asymptotically better than black-only pebbling, there are graphs exhibiting these separations for which optimal black-white pebblings can be simulated by bounded labelled pebblings. This means that resolution refutations of pebbling contradictions over such DAGs can do strictly asymptotically better than what is suggested by black-only pebbling, matching the bounds in terms of black-white pebbling.

More precisely, we such results can be obtained for three families of graphs. The first family are the *bit reversal graphs* studied in Section 8, for which Lengauer and Tarjan [LT82] established that black-white pebblings has quadratically better trade-offs than black pebblings. Recall that, more formally, they showed that there are DAGs $\{G_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ with black pebbling price $\mathsf{Peb}(G_n) = 3$ such that any optimal black pebbling $\mathcal{P}_n$ of G_n exhibits a trade-off $\mathsf{time}(\mathcal{P}_n) = \Theta\bigl(n^2/\mathsf{space}(\mathcal{P}_n) + n\bigr)$ but optimal black-white pebblings $\mathcal{P}_n$ of G_n achieve a trade-off $\mathsf{time}(\mathcal{P}_n) = \Theta\bigl((n/\mathsf{space}(\mathcal{P}_n))^2 + n\bigr)$.

Theorem 14.17 ([Nor10a]). *Fix any non-constant Boolean function f and let $Peb_{G_n}[f]$ be pebbling contradictions over the bit reversal graphs G_n of size $\Theta(n)$ in [LT82]. Then for any monotonically nondecreasing function $s(n) = \mathrm{O}(\sqrt{n})$ there are resolution refutations π_n of $Peb_{G_n}[f]$ in total space $\mathrm{O}(s(n))$ and length $\mathrm{O}\bigl(n^2/s(n)^2\bigr)$, beating the lower bound $\Omega\bigl(n^2/s(n)\bigr)$ for black-only pebblings of G_n.*

Focusing next on absolute bounds on space rather than time-space trade-offs, the best known separation between black and black-white pebbling for polynomial-size graphs is the one shown by Wilber [Wil88], who exhibited graphs $\{G(s)\}_{s=1}^{\infty}$ of size polynomial in s with black-white pebbling price $\mathsf{BW\text{-}Peb}(G(s)) = \mathrm{O}(s)$ and black pebbling price $\mathsf{Peb}(G(s)) = \Omega(s \log s/ \log\log s)$. For pebbling formulas over these graphs we do *not* know how to beat the black pebbling space bound—we return to this somewhat intriguing problem below—but using instead the graphs with essentially the same pebbling properties constructed in [KS91] and covered in Section 12, we can obtain the desired result.

Theorem 14.18 ([Nor10a]). *Fix any non-constant Boolean function f and let $Peb_{G(s)}[f]$ be pebbling contradictions over the graphs $G(s)$ in [KS91] with the same pebbling properties as in [Wil88]. Then there are resolution refutations π_n of $Peb_{G(s)}[f]$ in total space $\mathrm{O}(s)$, beating the lower bound $\Omega(s \log s/ \log\log s)$ for black-only pebbling.*

If we remove all restriction on graph size, there is a quadratic separation of black and black-white pebbling established by Kalyanasundaram and Schnitger [KS91]. Recall again from Section 12 that they proved that there are DAGs $\{G(s)\}_{s=1}^{\infty}$ of size $\exp(\Theta(s \log s))$ such that $\mathsf{BW\text{-}Peb}^{\bullet}(G(s)) \leq 3s + 1$ but $\mathsf{Peb}^{\bullet}(G(s)) \geq s^2$. For pebbling formulas over these graphs, resolution again matches the black-white pebbling bounds.

Theorem 14.19 ([Nor10a]). *Fix any non-constant Boolean function f and let $Peb_{G(s)}[f]$ be pebbling contradictions over the graphs $G(s)$ in [KS91] exhibiting a quadratic separation of black and black-white pebbling. Then there are resolution refutations π_n of $Peb_{G(s)}[f]$ in total space $\mathrm{O}(s)$, beating the lower bound $\Omega\bigl(s^2\bigr)$ for black-only pebbling.*

Note that, in particular, this means that if we want to prove *lower bounds* on resolution refutations of pebbling contradictions in terms of pebble games, the best we can hope for in general are bounds expressed in terms of black-white pebbling and not black-only pebbling.

Also, it should be noted that the best length-space separation that could possible be provided by pebbling contradictions are for formulas of size $\Theta(n)$ that are refutable in length $\mathrm{O}(n)$ but require clause space

$\Omega(n/\log n)$. This is so since as was discussed in Section 6, [HPV77] showed that any graph of size n with bounded maximal indegree has a black pebbling in space $\mathrm{O}(n/\log n)$. In fact, we can say more than that, namely that if any formula F has a resolution refutation π in length L, then it can be refuted in clause space $\mathrm{O}(L/\log L)$ (as was mentioned in Section 14.1.2). To see this, consider the graph representation G_π of π. By [HPV77], this graph can be black-pebbled in space $\mathrm{O}(L/\log L)$. It is not hard to see that we can construct another refutation that simulates this pebbling G_π by keeping exactly the clauses in memory that correspond to black-pebbled vertices, and that this refutation will preserve the pebbling space.[9]

In view of the results above, an intriguing open question is whether resolution can *always* simulate black-white pebblings, so that the refutation space of pebbling contradictions is asymptotically equal to the black-white pebbling price of the underlying graphs.

Open Problem 15 ([Nor10a]). *Is in true for any DAG G with bounded vertex indegree and any (fixed) Boolean function f that the pebbling contradiction $Peb_G[f]$ can be refuted in total space $\mathrm{O}(\textsf{BW-Peb}(G))$?*

More specifically, one could ask—as a natural first line of attack if one wants to investigate whether the answer to the above question could be yes—if it holds that bounded labelled pebblings are in fact as powerful as general black-white pebblings. In a sense, this is asking whether only a very limited form of nondeterminism is sufficient to realize the full potential of black-white pebbling.

Open Problem 16 ([Nor10a]). *Does it hold that any complete black-white pebbling $\mathcal{P}$ of a single-sink DAG G with bounded vertex indegree can be simulated by a $(\mathrm{O}(\textsf{space}(\mathcal{P})), \mathrm{O}(1))$-bounded pebbling $\mathcal{L}$?*

Note that a positive answer to this second question would immediately imply a positive answer to the first question as well by Lemma 14.16.

We have no strong intuition either way regarding Open Problem 15, but as to Open Problem 16 it would perhaps be somewhat surprising if bounded labelled pebblings turned out to be as strong as general black-white pebblings. Interestingly, although the optimal black-white pebblings of the graphs in [KS91] can be simulated by bounded pebblings, the same approach does *not* work for the original graphs separating black-white from black-only pebbling in [Wil88]. Indeed, these latter graphs might be a candidate graph family for answering Open Problem 16 in the negative, i.e., showing that standard black-white pebblings can be asymptotically stronger than bounded labelled pebblings.

14.4.2 Techniques for Lower Bounds on Resolution Trade-offs

To prove lower bounds on resolution refutations in terms of pebble games, we need to construct a reduction from refutations to pebblings. Let us again use formulas $Peb_G[\vee_2]$ to illustrate our reasoning.

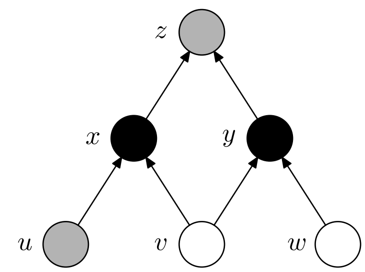
For black pebbles, we can reuse the ideas above for transforming pebblings into refutations and apply them in the other direction. That is, if the clause $v_1 \vee v_2$ is implied by the current content of the blackboard, we will let this correspond to a black pebble on v. A white pebble in a pebbling is a "debt" that has to be paid. It is difficult to see how any clause could be a liability in the same way and therefore no set of clauses corresponds naturally to isolated white pebbles. But if we think of white pebbles as assumptions that allow us to place black pebbles higher up in the DAG, it makes sense to say that if the content of the blackboard conditionally implies $v_1 \vee v_2$ given that we also assume the set of clauses $\{w_1 \vee w_2 \mid w \in W\}$ for some vertex set W, then this could be interpreted as a black pebble on v and white pebbles on the vertices in W.

Using this intuitive correspondence, we can translate sets of clauses in a refutation of $Peb_G[\vee_2]$ into black and white pebbles in G as in Figure 31. To see this, note that if we assume $v_1 \vee v_2$ and $w_1 \vee w_2$,

[9] As a matter of fact, the original definition of the clause space of a resolution refutation in [ET01] was as the black pebbling price of the graph G_π, but (the equivalent) Definition 14.2 as introduced by [ABRW02] has turned out to be more convenient to work with for most purposes.

$$\begin{bmatrix} x_1 \lor x_2 \\ \overline{v}_1 \lor \overline{w}_1 \lor y_1 \lor y_2 \\ \overline{v}_1 \lor \overline{w}_2 \lor y_1 \lor y_2 \\ \overline{v}_2 \lor \overline{w}_1 \lor y_1 \lor y_2 \\ \overline{v}_2 \lor \overline{w}_2 \lor y_1 \lor y_2 \end{bmatrix}$$

(a) Clauses on blackboard.



(b) Corresponding pebbles in the graph.

Figure 31: Intuitive translation of clauses to black and white pebbles.

this assumption together with the clauses on the blackboard in Figure 31(a) imply $y_1 \lor y_2$, so y should be black-pebbled and v and w white-pebbled in Figure 31(b). The vertex x is also black since $x_1 \lor x_2$ certainly is implied by the blackboard. This translation from clauses to pebbles is arguably quite straightforward, and furthermore it seems to yield well-behaved black-white pebblings for all "sensible" resolution refutations of $Peb_G[\lor_2]$. (What this actually means is that all refutations of pebbling contradictions that we are able to come up with can be described as simulations of labelled pebblings as defined in Definition 14.13, and for such refutations the reduction just sketched will essentially give us back the pebbling we started with.)

The problem, however, is that we have no guarantee that 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. It can happen that clauses are derived which cannot be translated, at least not in a natural way, to pebbles in the fashion indicated above.

Some of these clauses we can afford to ignore. For example, considering how axiom clauses can be used in derivations it seems reasonable to expect that a derivation never writes an isolated axiom $\overline{v}_i \lor \overline{w}_j \lor y_1 \lor y_2$ on the blackboard. And in fact, if three of the four axioms for v in Figure 31 are written on the blackboard but the fourth one $\overline{v}_2 \lor \overline{w}_2 \lor y_1 \lor y_2$ is missing, we will just discard these three clauses and there will be no pebbles on v, w, and y corresponding to them.

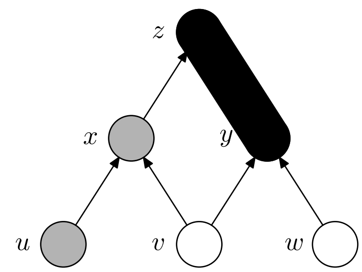
A more dangerous situation is when clauses are derived that are the disjunction of positive literals from different vertices. For instance, a derivation starting from Figure 31(a), could add the axioms $\overline{x}_1 \lor \overline{y}_2 \lor z_1 \lor z_2$ and $\overline{x}_2 \lor \overline{y}_2 \lor z_1 \lor z_2$ to the blackboard, derive that the truth of v and w implies the truth of either y or z, i.e., the clauses $\overline{v}_i \lor \overline{w}_j \lor y_1 \lor z_1 \lor z_2$ for $i, j = 1, 2$, and then erase $x_1 \lor x_2$ to save space, resulting in the blackboard in Figure 32(a). As it stands, the content of this blackboard does not correspond to any pebbles under our tentative translation. However, the clauses can easily be used to derive something that does. For instance, writing down all axioms $\overline{x}_i \lor \overline{y}_j \lor z_1 \lor z_2$, $i, j = 1, 2$, on the blackboard, we get that the truth of v, w, and x implies the truth of z. We have decided to interpret such a set of clauses as a black pebble on z and white pebbles on v, w, and x, but this pebble configuration cannot arise out of nothing in an empty DAG. Hence, the clauses in Figure 32(a) have to correspond to some set of pebbles. But what pebbles?

Although it is hard to motivate from such a small example, this turns out to be a very serious problem. There appears to be no way that we can interpret derivations as the one described above in terms of black and white pebbles without making some component in the reduction from resolution to pebbling break down.

So what can we do? Well, if you can't beat 'em, join 'em! In order to prove their results, [Nor09a, NH08b, BN08] gave up the attempts to translate resolution refutations into black-white pebblings and instead invented new pebble games (in three different flavours). These pebble games are on the one hand somewhat similar to the black-white pebble game, but on the other hand they have pebbling rules specifically designed so that tricky clause sets such as that in Figure 32(a) can be interpreted in a satisfying

$$\begin{bmatrix} \overline{v}_1 \lor \overline{w}_1 \lor y_1 \lor z_1 \lor z_2 \\ \overline{v}_1 \lor \overline{w}_2 \lor y_1 \lor z_1 \lor z_2 \\ \overline{v}_2 \lor \overline{w}_1 \lor y_1 \lor z_1 \lor z_2 \\ \overline{v}_2 \lor \overline{w}_2 \lor y_1 \lor z_1 \lor z_2 \end{bmatrix}$$

(a) New set of clauses on blackboard.



(b) Corresponding blobs and pebbles.

Figure 32: Intepreting sets of clauses as black blobs and white pebbles.

fashion. Once this has been taken care of, one proceeds with the construction of the proof as outlined in Section 14.3.1, but using the modified pebble games instead of standard black-white pebbling. In what follows, we describe how this is done employing the pebble game defined in [BN08] (though using the more evocative terminology from [NH08b]). The games in [Nor09a, NH08b], although somewhat different on the surface, can also be recast in the framework presented below.

The new pebble game in [BN08] is similar to the one in Definition 14.13, but with a crucial change in the definition of the "subconfigurations." There are white pebbles just as before, but the black pebbles are generalized to *blobs* that can cover multiple vertices instead of just a single vertex. A blob on a vertex set V can be thought of as truth of some vertex $v \in V$, unknown which. The clauses in Figure 32(a) are consequently translated into white pebbles on v and w, as before, and a black blob covering both y and z as in Figure 32(b). To parse the formal definition of the game, it might be helpful to look at Figure 33.

Definition 14.20 (Blob-pebble game [BN08]). If B and W are sets of vertices with $B \neq \emptyset$, $B \cap W = \emptyset$, we say that $[B]\langle W \rangle$ is a *blob subconfiguration* with black pebbles on B and white pebbles on W *supporting* B. A *blob-pebbling* of a DAG G with unique sink z is a sequence $\mathcal{P} = \{\mathbb{P}_0, \ldots, \mathbb{P}_\tau\}$ of sets of blob subconfigurations, or *blob-pebbling configurations*, such that $\mathbb{P}_0 = \emptyset$, $\mathbb{P}_\tau = \{[z]\langle\emptyset\rangle\}$, and for all $t \in [\tau]$, $\mathbb{P}_t$ is obtained from $\mathbb{P}_{t-1}$ by one of the following rules:

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

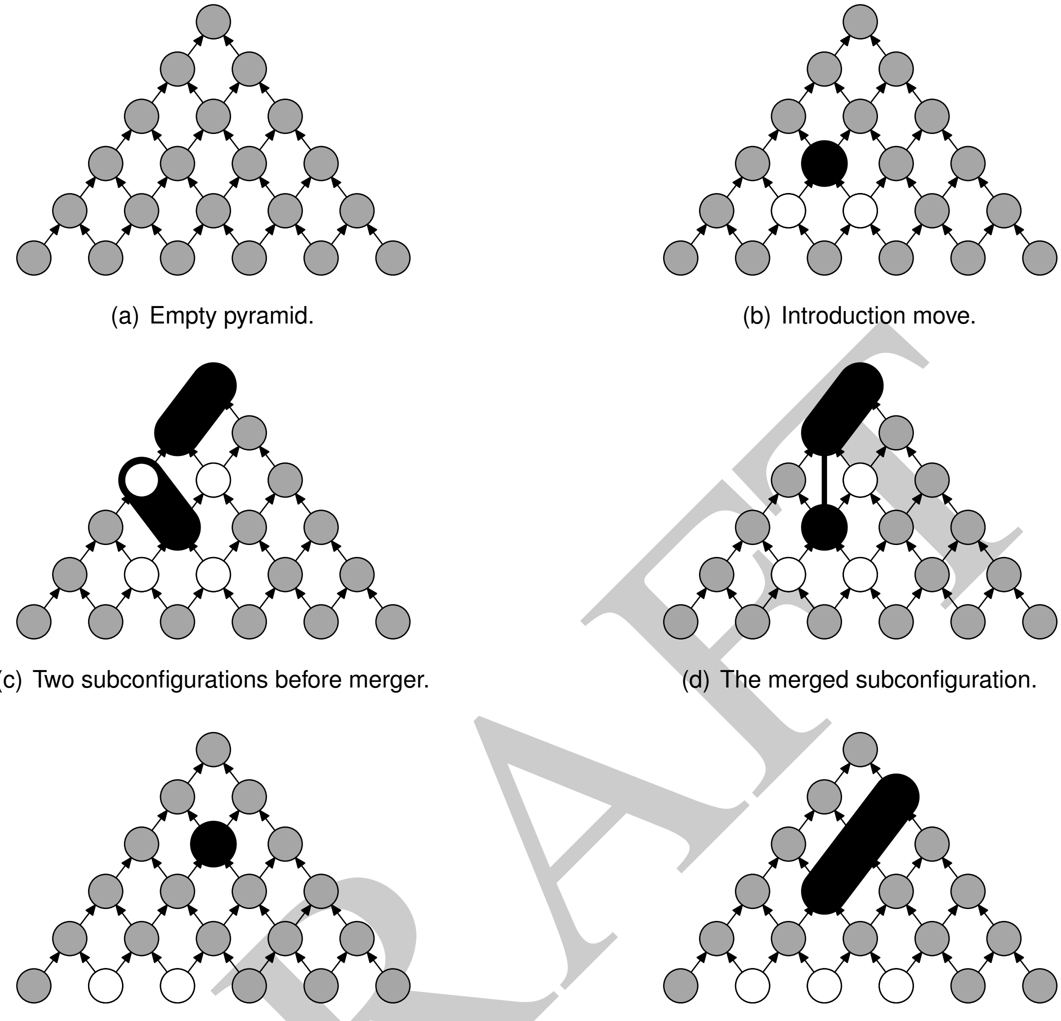
Merger $\mathbb{P}_t = \mathbb{P}_{t-1} \cup \{[B_1 \cup B_2]\langle W_1 \cup W_2\rangle\}$ if there are $[B_1]\langle W_1 \cup \{v\}\rangle, [B_2 \cup \{v\}]\langle W_2\rangle \in \mathbb{P}_{t-1}$ such that $B_1 \cap W_2 = \emptyset$.

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

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

Let us now return to the proof outline in Section 14.3.1. The first step in our approach is to establish that any resolution refutation of a pebbling contradiction can be interpreted as a pebbling (but now in our modified game) of the DAG in terms of which this pebbling contradiction is defined. Intuitively, axiom downloads in the refutation will be matched by introduction moves in the blob-pebbling, erasures correspond to erasures, and seemingly suboptimal derivation steps can be modelled by inflation moves in the blob-pebbling. In all three papers [Nor09a, NH08b, BN08], the formal definitions are set up so that a theorem along the following lines can be proven.

Tentative Theorem 14.21. *Consider a pebbling contradiction $Peb_G[f]$ over any DAG G. Then there is a translation function from sets of clauses over $Vars(Peb_G[f])$ to sets of black blobs and white pebbles in G that translates any resolution refutation π of $Peb_G[f]$ into a blob-pebbling $\mathcal{P}_\pi$ of G.*



(a) Empty pyramid.
(b) Introduction move.
(c) Two subconfigurations before merger.
(d) The merged subconfiguration.
(e) Subconfiguration before inflation.
(f) Subconfiguration after inflation.

Figure 33: Examples of moves in the blob-pebble game.

The next step is to show pebbling lower bounds. Since the rules in the blob-pebble game are different from those of the standard black-white pebble game, known bounds on black-white pebbling price in the literature no longer apply. But again, provided that we have got the right definitions in place, we hope to be able to establish that the blob-pebblings can do no better than standard black-white pebblings.

Tentative Theorem 14.22. *If there is a blob-pebbling of a DAG G in time τ and space s, then there is a standard black-white pebbling of G in time $\mathrm{O}(\tau)$ and space $\mathrm{O}(s)$.*

Finally, we need to establish that the number of pebbles used in $\mathcal{P}_\pi$ in Tentative Theorem 14.21 is related to the space of the resolution refutation π. As we know from Section 14.3.1, such a bound cannot be true for formulas Peb_G so this is where we need to do substitutions with some suitable Boolean function f_d over $d \geq 2$ variables and study $Peb_G[f_d]$.

Tentative Theorem 14.23. *If π is a resolution refutation of a pebbling contradiction $Peb_G[f_d]$ for some suitable Boolean function f_d, then the time and space of the associated blob-pebbling $\mathcal{P}_\pi$ of G are upper bounded by π by $\textsf{time}(\mathcal{P}_\pi) = \mathrm{O}(L(\pi))$ and $\textsf{space}(\mathcal{P}_\pi) = \mathrm{O}(Sp(\pi))$.*

If we put these three theorems together, it is clear that we can translate pebbling trade-offs to resolution trade-offs as described in the "theorem template" at the end of Section 14.3.1.

There is a catch, however, which is why we have used the label "tentative theorems" above. It is reasonably straightforward to come up with natural definitions that allow us to prove Tentative Theorem 14.21. But this in itself does not yield any lower bounds. (Indeed, there is a natural translation from refutations to pebbling even for Peb_G, for which we know that the lower bounds we are after do *not* hold!) The lower bounds instead follow from the combination of Tentative Theorems 14.22 and 14.23, but there is a tension between these two theorems.

The attentive reader might already have noted that two crucial details in Definition 14.20 are missing—we have not defined pebbling time and space for blob-pebblings. And for a good reason, because this turns out to be where the difficulty lies. On the one hand, we want the time and space measures for blob-pebblings to be as strong as possible, so that we can make Tentative Theorem 14.22 hold, saying that blob-pebblings are no stronger than standard pebblings. On the other hand, we do not want the definitions to be too strong, for if so the bounds we need in Tentative Theorem 14.23 might break down. This turns out to be the major technical difficulty in the construction

In the papers [Nor09a, NH08b], which study formulas $Peb_G[\vee_2]$ defined in terms of binary logical or, we cannot make any connection between pebbling time and refutation length in Tentative Theorems 14.22 and 14.23, but instead have to focus on only clause space. Also, the constructions work not for general DAGs but only for binary trees in [Nor09a], and only for a somewhat more general class of graphs also including pyramids in [NH08b]. The reason for this is that it is hard to charge for black blobs and white pebbles. If we could charge for all vertices covered by blobs and pebbles, or at least one space unit for every black blob and every white pebble, we would be in good shape. But it appears hard to do so without losing the connection to clause space that we want in Tentative Theorem 14.23. Instead, for formulas $Peb_G[\vee_2]$ the best space measure that we can come up with is as follows.

Definition 14.24 (Blob-pebbling price with respect to $\vee_2$). Let $\mathbb{P} = \{[B_i]\langle W_i\rangle \mid i = 1, \ldots, n\}$ be a set of blob subconfigurations over some DAG G.

A *chargeable black blob collection* of $\mathbb{P}$ is an ordered subset $\{B_1, \ldots, B_m\}$ of black blobs in $\mathbb{P}$ such that for all $i \leq m$ it holds that $B_i \setminus \bigcup_{j<i} B_j \neq \emptyset$ (i.e., the unions $\bigcup_{j<i} B_j$ are strictly expanding for $i = 1, \ldots, m$). We say that such a collection has *black cost* m.

The set of *chargeable white pebbles* of a subconfiguration $[B_i]\langle W_i\rangle \in \mathbb{P}$ is the subset of vertices $w \in W_i$ that are below all $b \in B_i$ (where "below" means that there is a path from w to b in G). The *chargeable white pebble collection* of $\mathbb{P}$ is the union of all such vertices for all $[B_i]\langle W_i\rangle \in \mathbb{P}$, and the *white cost* is the size of this set.

The *space* of a blob-pebbling configuration $\mathbb{P}$ is the largest black cost of a chargeable blob collection plus the cost of the chargeable white pebble collection, and the space of a blob-pebbling is the maximal space of any blob-pebbling configuration in it. The blob-pebbling price *Blob-Peb*(G) of a DAG G is the minimum space of any complete blob-pebbling of G.

Using the translation of clauses to blobs and pebbles in [BN08] it can be verified that Tentative Theorem 14.21 as proven in that paper holds also for formulas $Peb_G[\vee_2]$. Moreover, extending the proof techniques in [Nor09a, NH08b] it is also not too hard to show the space bound in Tentative Theorem 14.23.[10] But we do not know how to establish the space part in Tentative Theorem 14.22 for general DAGs. This

[10] Although it is phrased in very different terms, what is shown in [Nor09a, NH08b] is essentially the somewhat more restricted result that if we charge only for the set of black vertices V such that every $v \in V$ is the unique bottom black vertex in some subconfiguration $[B]\langle W\rangle$ that have all vertices $b \in B$ topologically ordered (i.e., the blob B is a chain) and only for supporting white pebbles $w \in W$ that are located below their bottom black vertex in such subconfigurations, then the space bound in Tentative Theorem 14.23. holds. The proofs in [Nor09a, NH08b] extend to the more general definition of blob-pebbling space in Definition 14.24, however.

is the part of the construction where [Nor09a] works only for binary trees and [NH08b] works only for pyramids and friends.

The crucial new idea in [BN08] to make the approach outlined above work for general DAGs was to switch formulas from $Peb_G[\vee_2]$ to $Peb_G[f]$ for other functions f, such as for instance binary exclusive or $\oplus_2$. However, while this does make the analysis much simpler (and stronger), it is not at all clear that the change of formulas should be necessary. We find it an intriguing question whether the program in Tentative Theorems 14.21, 14.22, and 14.23 could in fact be carried out for formulas $Peb_G[\vee_2]$.

Open Problem 17 ([Nor10a]). *Is it true for any DAG G that any resolution refutation π of $Peb_G[\vee_2]$ can be translated into a black-white pebbling of G with time and space upper-bounded in terms of the length and space of π?*

In particular, can we translate upper bounds in the blob-pebble game in Definition 14.20 with space defined as in Definition 14.24 to upper bounds for standard black-white pebbling? (From which clause space lower bounds for $Peb_G[\vee_2]$ would immediately follow.)

Our take on the results in [Nor09a, NH08b] is that they can be interpreted as indicating that this should indeed be the case. Although, as noted above, these results only apply to limited classes of graphs, and only capture space lower bounds and not time-space trade-offs, the problems arising in the analysis seem to have to do more with artifacts in the proofs than with any fundamental differences between formulas $Peb_G[\vee_2]$ and, say, $Peb_G[\oplus_2]$. We remark that the papers [BN08, BN11] do not shed any light on this question, as the techniques used there inherently cannot work for formulas defined in terms of non-exclusive logical or.

If Open Problem 17 could be resolved in the positive, this could potentially be useful for settling the complexity of decision problems for resolution proof space, i.e., the problem given a CNF formula F and a space bound s to determine whether F has a resolution refutation in space at most s. Reducing from pebbling space by way of formulas $Peb_G[\vee_2]$ would avoid the blow-up of the gap between upper and lower bounds on pebbling space that cause problems when using, for instance, exclusive or.

But let us return to the paper [BN08] that resolves the problems identified in [Nor09a, NH08b]. The reason that we gain from switching from formulas $Peb_G[\vee_2]$ to, for instance, formulas $Peb_G[\oplus_2]$ is that for the latter formulas we can define a much stronger space measure for the blob-pebblings. In this case, it turns out that one can in fact charge for all vertices covered by blobs or pebbles in the blob-pebble game, and then the space bound in Tentative Theorem 14.23 follows for arbitrary DAGs. In the follow-up work [BN11] this result was improved to capture not only space but also the connection between pebbling time and refutation length, thus realizing the full program described in Section 14.3.1.

In this process, [BN11] also presented a much cleaner way to argue more generally about how the refutation length and space of a CNF formula F change when we do substitution with some Boolean function f to obtain $F[f]$. Unfortunately, describing this technique in more detail cannot be done within the scope of this survey, and we instead refer to [Nor10b] for more details. However, we conclude this discussion by giving some examples from [BN08, BN11] of the kind of results obtained by this method.

14.4.3 Statement of Space Lower Bounds and Length-Space Trade-offs

Regarding the question of the relationship between length and space in resolution, [BN08] showed that in contrast to the tight relation between length versus width, length and space are as uncorrelated as they can possibly be.

Theorem 14.25 (Length-space separation for resolution [BN08]). *There exist explicitly constructible families of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ that can be refuted in resolution in length $\mathrm{O}(n)$ and width $\mathrm{O}(1)$ simultaneously, but for which any resolution refutation must have clause space $\Omega(n/\log n)$.*

An extension of this theorem to k-DNF resolution in [BN11] showed that this family of proof systems does indeed form a strict hierarchy with respect to space.

Theorem 14.26 (k-DNF resolution space hierarchy [BN11]). *For every $k \geq 1$ there exists an explicitly constructible family $\{F_n\}_{n=1}^{\infty}$ of CNF formulas of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that there are $\mathcal{R}(k+1)$-refutations $\pi_n : F_n \vdash 0$ in simultaneous length $L(\pi_n) = \mathrm{O}(n)$ and formula space $Sp(\pi_n) = \mathrm{O}(1)$, but any $\mathcal{R}(k)$-refutation of F_n requires formula space $\Omega\bigl(\sqrt[k+1]{n/\log n}\bigr)$. The constants hidden by the asymptotic notation depend only on k.*

The formula families $\{F_n\}_{n=1}^{\infty}$ in Theorems 14.25 and 14.26 are obtained by considering pebbling contradictions over the graphs examined in Section 7 requiring black-white pebbling space $\Theta(n/\log n)$, and substituting a k-non-authoritarian Boolean function f of arity $k+1$, for instance XOR over $k+1$ variables, in these formulas.

The above theorems give absolute lower bounds on space for resolution and $\mathcal{R}(k)$. Combining the techniques in [BN11] we can also derive length-space trade-offs for these proof systems. In fact, we can obtain a multitude of such trade-offs, since for any graph family with tight dual trade-offs for black and black-white pebbling, or for which black-white pebblings can be cast in the framework of Section 14.4.1 and simulated by resolution, we can obtain a corresponding trade-off for resolution-based proof systems. Since a full catalogue listing all of these trade-off results would be completely unreadable, we try to focus on some of the more salient examples below.

From the point of view of space complexity, the easiest formulas are those refutable in constant total space, i.e., formulas having so simple a structure that there are resolution refutations where we never need to keep more than a constant number of symbols on the proof blackboard. A priori, it is not even clear whether we should expect that any trade-off phenomena could occur for such formulas, but it turns out that there are quadratic length-space trade-offs.

Theorem 14.27 (Quadratic trade-offs for constant space [BN11]). *For any fixed positive integer K there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that the following holds (where all multiplicative constants hidden in the asymptotic notation depend only on K):*

- *The formulas F_n are refutable in syntactic resolution in total space $TotSp_{\mathcal{R}}(F_n \vdash 0) = \mathrm{O}(1)$.*
- *For any monotone function $s_{\mathrm{hi}}(n) = \mathrm{O}\bigl(\sqrt{n}\bigr)$ there are syntactic resolution refutations π_n of F_n in simultaneous length $L(\pi_n) = \mathrm{O}\bigl((n/s_{\mathrm{hi}}(n))^2\bigr)$ and total space $TotSp(\pi_n) = \mathrm{O}(s_{\mathrm{hi}}(n))$.*
- *For any semantic resolution refutation $\pi_n : F_n \vdash 0$ in clause space $Sp(\pi_n) \leq s_{\mathrm{hi}}(n)$ it holds that $L(\pi_n) = \Omega\bigl((n/s_{\mathrm{hi}}(n))^2\bigr)$.*
- *For any $k \leq K$, any semantic k-DNF resolution refutation π_n of F_n in formula space $Sp(\pi_n) \leq s_{\mathrm{hi}}(n)$ must have length $L(\pi_n) = \Omega\Bigl(\bigl(n/\bigl(s_{\mathrm{hi}}(n)^{1/(k+1)}\bigr)\bigr)^2\Bigr)$. In particular, any constant-space $\mathcal{R}(k)$-refutation must also have quadratic length.*

Theorem 14.27 follows by combining the pebbling trade-offs in Section 8 with the structural results on simulations of black-white pebblings by resolution in Theorem 14.17.

Remark 14.28. Notice that the trade-off applies to both formula space—i.e., clause space for $\mathcal{R}(1)$—and total space. This is because the upper bound is stated in terms of the larger of these two measures (total space) while the lower bound is in terms of the smaller one (formula space). Note also that the upper bounds hold for the usual, syntactic versions of the proof systems, whereas the lower bounds hold for the much stronger semantic systems, and that for standard resolution the upper and lower bounds are tight up to constant factors. These properties hold for all trade-offs stated below. Finally, we remark that we have to pick some arbitrary but fixed limit K for the size of the terms when stating the results for k-DNF resolution, since for any family of formulas we consider there will be very length- and space-efficient $\mathcal{R}(k)$-refutation refutations if we allow terms of unbounded size.

Our next example is based on the pebbling trade-off result in Section 9. Using this new result, we can derive among other things the rather striking statement that for any *arbitrarily slowly growing* non-constant function, there are explicit formulas of such (arbitrarily small) space complexity that nevertheless exhibit *superpolynomial* length-space trade-offs.

Theorem 14.29 (Superpolynomial trade-offs for arbitrarily slowly growing space [BN11]). *Let $s_{\mathrm{lo}}(n) = \omega(1)$ be any arbitrarily slowly growing function*[11] *and fix any $\epsilon > 0$ and positive integer K. Then there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that the following holds:*

- *The formulas F_n are refutable in syntactic resolution in total space $\mathit{TotSp}_{\mathcal{R}}(F_n \vdash 0) = \mathrm{O}(s_{\mathrm{lo}}(n))$.*
- *There are syntactic resolution refutations π_n of F_n in simultaneous length $L(\pi_n) = \mathrm{O}(n)$ and total space $\mathit{TotSp}(\pi_n) = \mathrm{O}\left(\left(n/s_{\mathrm{lo}}(n)^2\right)^{1/3}\right)$.*
- *Any semantic resolution refutation of F_n in clause space $\mathrm{O}\left(\left(n/s_{\mathrm{lo}}(n)^2\right)^{1/3-\epsilon}\right)$ must have superpolynomial length.*
- *For any $k \leq K$, any semantic $\mathcal{R}(k)$-refutation of F_n in formula space $\mathrm{O}\left(\left(n/s_{\mathrm{lo}}(n)^2\right)^{1/(3(k+1))-\epsilon}\right)$ must have superpolynomial length.*

All multiplicative constants hidden in the asymptotic notation depend only on K, ϵ and s_{lo}.

Observe the robust nature of this trade-off, which is displayed by the long range of space complexity in standard resolution, from $\omega(1)$ up to $\approx n^{1/3}$, which requires superpolynomial length. Note also that the trade-off result for standard resolution is very nearly tight in the sense that the superpolynomial lower bound on length in terms of space reaches up to very close to where the linear upper bound kicks in.

The two theorems above focus on trade-offs for formulas of low space complexity, and the lower bounds on length obtained in the trade-offs are somewhat weak—the superpolynomial growth in Theorem 14.29 is something like $n^{s_{\mathrm{lo}}(n)}$. We next present a theorem that has both a stronger superpolynomial length lower bounds than Theorem 14.29 and an even more robust trade-off covering a wider (although non-overlapping) space interval. This theorem again follows by applying our tools to the pebbling trade-offs in Section 10 established by [LT82].

Theorem 14.30 (Robust superpolynomial trade-off for medium-range space [BN11]). *For any positive integer K, there are explicitly constructible CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ and width $\mathrm{O}(1)$ such that the following holds (where the hidden constants depend only on K):*

- *The formulas F_n are refutable in syntactic resolution in total space $\mathit{TotSp}_{\mathcal{R}}(F_n \vdash 0) = \mathrm{O}(\log^2 n)$.*
- *There are syntactic resolution refutations of F_n in length $\mathrm{O}(n)$ and total space $\mathrm{O}(n/\log n)$.*
- *Any semantic resolution refutation of F_n in clause space $\mathit{Sp}(\pi_n) = \mathrm{o}(n/\log n)$ must have length $L(\pi_n) = n^{\Omega(\log \log n)}$.*
- *For any $k \leq K$, any semantic $\mathcal{R}(k)$-refutation in formula space $\mathit{Sp}(\pi_n) = \mathrm{o}\left((n/\log n)^{1/(k+1)}\right)$ must have length $L(\pi_n) = n^{\Omega(\log \log n)}$.*

[11]For technical reasons, we also assume that $s_{\mathrm{lo}}(n) = \mathrm{O}(n^{1/7})$, i.e., that $s_{\mathrm{lo}}(n)$ does not grow too quickly. This restriction is inconsequential since for faster-growing functions the trade-off results presented below yield much stronger bounds.

Having presented trade-off results in the low-space and medium-space range, we conclude by presenting a result at the other end of the space spectrum. Namely, appealing one last time to yet another result in [LT82], this time from Section 11, we can deduce that there are formulas of nearly linear space complexity (recall that any formula is refutable in linear formula space) that exhibit not only superpolynomial but even exponential trade-offs.

We state this final theorem for standard resolution only since it is not clear whether it makes sense for $\mathcal{R}(k)$. That is, we can certainly derive formal trade-off bounds in terms of the $(k+1)$st square root as in the theorems above, but we do not know whether there actually exist $\mathcal{R}(k)$-refutation in sufficiently small space so that the trade-offs apply. Hence, such trade-off claims for $\mathcal{R}(k)$, although impressive-looking, might simply be vacuous. It is possible to obtain other exponential trade-offs for $\mathcal{R}(k)$ but they are not quite as strong as the result below for resolution. We refer to [BN11] for the details.

Theorem 14.31 (Exponential trade-offs for nearly-linear space [BN11]). *Let κ be any sufficiently large constant. Then there are CNF formulas F_n of size $\Theta(n)$ and width $\mathrm{O}(1)$ and a constant $\kappa' \ll \kappa$ such that:*

- *The formulas F_n have syntactic resolution refutations in total space $\kappa' \cdot n / \log n$.*
- *F_n is also refutable in syntactic resolution in length $\mathrm{O}(n)$ and total space $\mathrm{O}(n)$ simultaneously.*
- *However, any semantic refutation of F_n in clause space at most $\kappa \cdot n / \log n$ has length $\exp\bigl(n^{\Omega(1)}\bigr)$.*

To get a feeling for this last trade-off result, note again that the lower bound holds for proof systems with arbitrarily strong derivation rules, as long as they operate with disjunctive clauses. In particular, it holds for proof systems that can in one step derive anything that is semantically implied by the current content of the blackboard. Recall that such a proof system can refute any unsatisfiable CNF formula F with n clauses in length $n+1$ simply by writing down all clauses of F on the blackboard and then concluding, in one single derivation step, the contradictory empty clause implied by F. In Theorem 14.31 the semantic resolution proof system has space nearly sufficient for such an ultra-short refutation of the whole formula. But even so, when we feed this proof system the formulas F_n and restrict it to having at most $\mathrm{O}(n/\log n)$ clauses on the blackboard at any one given time, it will have to keep going for an exponential number of steps before it is finished.

Anticipating the theme of the next subsection, we conclude our overview of time-space trade-off results for resolution-based proof system with the following intriguing question.

Open Problem 18. *Is it possible to develop the method in [BN11] further to prove trade-offs between proof length/size and space for Cutting Planes, Polynomial Calculus, or Polynomial Calculus with Resolution?*

One can find examples showing that the particular techniques in [BN11] will provably not work for Cutting Planes, Polynomial Calculus, or Polynomial Calculus with Resolution. However, it does not seem impossible that the approach in [BN11] can be strengthened to get around these counter-examples, but this in turn leads to additional technical difficulties in the analysis. We refer to [Nor10b] for a somewhat more detailed discussion of these issues.

14.5 Some Open Problems Regarding Space Bounds and Trade-offs

Despite the progress made during the last few years on understanding space in resolution and how it is related to other measures, there are quite a few natural questions that still have not been resolved.

Perhaps one of the main open questions is how complex a k-CNF formula F can be with respect to total space. If F has at most n clauses or variables (which is the case if, in particular, F has size n) we know from [ET01] that $Sp_{\mathcal{R}}(F \vdash 0) \leq n + \mathrm{O}(1)$. From this it immediately follows that $TotSp_{\mathcal{R}}(F \vdash 0) = \mathrm{O}\bigl(n^2\bigr)$. But it this upper bound tight?

Open Problem 19 ([ABRW02]). *Are there k-CNF formula families $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that $\mathit{TotSp}(F_n \vdash 0) = \Omega(n^2)$?*

As a first step towards resolving this question, Alekhnovich et al. [ABRW02] posed the problem of finding k-CNF formulas over n variables and of size polynomial in n such that $\mathit{TotSp}(F \vdash 0) = \omega(n)$. (There is a lower bound $\mathit{TotSp}(F \vdash 0) = \Omega(n^2)$ proven in [ABRW02], but it is for formulas of exponential size and linear width). Alekhnovich et al. also conjectured that there do exist formulas F_n of size n such that $\mathit{TotSp}(F_n \vdash 0) = \Omega(n^2)$, and suggested so-called Tseitin formulas defined over 3-regular expander graphs as a candidate for proving this.

The next two questions that we want to address concern upper bounds on resolution length in terms of clause space. We know from [AD08] that clause space is an upper bound for width, and width yields an upper bound on length by a simple counting argument. However, it would be more satisfying to gain a more direct understanding of *why* clause space upper-bounds length. Focusing on constant clause space for concreteness, the problem can be formulated as follows.

Open Problem 20. *For k-CNF formulas F of size n, we know that $\mathit{Sp}(F \vdash 0) = \mathrm{O}(1)$ implies $L(F \vdash 0) = \mathrm{poly}(n)$. Is there a direct proof of this fact, not going via [AD08]?*

If we could understand this problem better, we could perhaps also find out whether it is possible to derive stronger upper bounds on length in terms of space. Esteban and Torán ask the following question.

Open Problem 21 ([ET01]). *Does it hold for k-CNF formulas F that $\mathit{Sp}(F \vdash 0) = \mathrm{O}(\log n)$ implies $L(F \vdash 0) = \mathrm{poly}(n)$?*

Turning to the relationship between width and length, recall that we know from [BW01] that short resolution refutations imply the existence of narrow refutations, and that in view of this an appealing proof search heuristic is to search exhaustively for refutations in minimal width. One serious drawback of this approach, however, is that there is no guarantee that the short and narrow refutations are the same one. On the contrary, the narrow refutation constructed in the proof in [BW01] is potentially exponentially longer than the short refutation that one starts 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 interesting to know whether this increase in length is necessary. That is, is there a formula family which exhibits a length-width trade-off in the sense that there are short refutations and narrow refutations, but all narrow refutations have a length blow-up (polynomial or superpolynomial)? Or is the exponential blow-up in [BW01] just an artifact of the proof?

Open Problem 22 ([NH08b]). *If F is a k-CNF formula over n variables refutable in length L, can one always find a refutation π of F in width $W(\pi) = \mathrm{O}\big(\sqrt{n \log L}\big)$ with length no more than, say, $L(\pi) = \mathrm{O}(L)$ or at most $\mathrm{poly}(L)$? Or is there a formula family which necessarily exhibits a length-width trade-off in the sense that there are short refutations and narrow refutations, but all narrow refutations have a length blow-up (polynomial or superpolynomial)?*

As was mentioned above, for tree-like resolution Ben-Sasson [Ben09] showed that there are formulas F_n refutable in linear length and also in constant width, but for which any refutation π_n must have $W(\pi_n) \cdot \log L(\pi_n) = \Omega(n/\log n)$. This shows that the length blow-up in the proof of the tree-like length-width relationships in [BW01] is necessary. That is, transforming a short tree-like proof into a narrow proof might necessarily incur an exponential length blow-up. But tree-like resolution is very different from unrestricted resolution in that upper bounds on width do *not* imply upper bounds on length (as shown in [BW01] using $\mathit{Peb}_G[\vee_2]$-formulas), so it is not clear that the result for tree-like resolution provides any intuition for the general case.

A related question about trade-offs between length and width on a finer scale, raised by Albert Atserias and Marc Thurley, is as follows.

Open Problem 23 ([AT09]). *For $w \geq 3$ arbitrary but fixed, is there family of unsatisfiable 3-CNF formulas $\{F_n^w\}_{n=1}^{\infty}$ of size $\Theta(n)$ that have resolution refutations of width w but cannot be refuted in length $\mathrm{O}\big(n^{w-c}\big)$ for some small positive constant c?*

This question was prompted by the paper [AFT09], where it was shown that there are SAT solvers which can refute formulas F_n with $W(F_n \vdash 0) = w$ in time roughly n^{2w}. It is natural to ask how much room for improvement there is for this time bound. Since these algorithms are resolution-based, it would be nice if one could prove a lower bound saying that there are formulas F_n with $W(F_n \vdash 0) = w$ that cannot be refuted by resolution in length $n^{\mathrm{o}(w)}$, or even $n^{w-\mathrm{O}(1)}$. As a step towards proving (or disproving) this, resolving special cases of Open Problem 23 for concrete instantiations of the parameters, say $w = 10$ and $w - c = 2$, would also be of interest.

For resolution clause space, we know that there can be very strong trade-offs with respect to length for space s in the range $\omega(1) = s = \mathrm{o}(n/\log\log n)$, but we do not know what holds for space outside of this range. Consider first formulas refutable by proofs π in constant space. When we run such a refutation through the proof in [AD08] and obtain another narrow, and thus short, refutation π' we do not have any information about the space complexity of this refutation. Is it possible to get a refutation in both short length and small space simultaneously?

Open Problem 24 ([NH08b]). *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 resolution 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?*

Note that if we instead look at the total space measure (that also counts the number of literals in each clause with repetitions), then the answer to the above question is that we can obtain refutations that are both short and space-efficient simultaneously, again by a simple counting argument. But for clause space such a counting argument does not seem to apply, and maybe strange things can happen. (They certainly can in the sense that as soon as we go to arbitrarily slowly growing non-constant space, there provably exist strong space-length trade-offs.)

For concreteness, fix space to the bare minimum and suppose for a k-CNF formula F of size n that $Sp(F \vdash 0) = 3$. Does this imply that there is a resolution refutation $\pi : F \vdash 0$ in simultaneous length $L(\pi) = \mathrm{poly}(n)$ and $Sp(\pi) = 3$? (Or at least $Sp(\pi) = \mathrm{O}(1)$?) Observe that if one looks at the graph representation G_π of a space 3 refutation π it has a very special structure—it is just a long line with each vertex on the line having indegree 2 and its non-line predecessor being an axiom clause. This is so since all such a refutation can do is download an axiom, resolve with the clause currently in memory, keep the resolvent and erase the two clauses just resolved. The question is now whether it can be the case that a refutation with such a simple graph structure can have to go on for a superpolynomial number of steps. Even this simple question is open. Of course, one would expect here that any insight regarding Open Problem 20 should have bearing on this question as well.

Consider now space complexity at the other end of the range. Note that all trade-offs for clause space proven so far are in the regime where the space $Sp(\pi)$ is less than the number of clauses $|F|$ in F. On the one hand, this is quite natural, since the size of the formula is an upper bound on the refutation clause space needed. On the other hand, it is not clear that this should rule out length-space trade-offs for linear or superlinear space, since the proof that any formula is refutable in linear space constructs a resolution refutation that has exponential length. Assume therefore that we have a CNF formula F of size 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 25 ([Ben07, NH08b]). *Let F be any k-CNF formula with $|F| = n$ clauses. 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 $L(\pi) = \mathrm{poly}(L)$? Or are there formulas with trade-offs in the range space $\geq$ formula size?*

If so, this could be interpreted as saying that a smart enough clause learning algorithm could 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).

Finally, a slightly curious aspect of the space lower bounds and length-space trade-offs surveyed above is that the results in [Nor09a, NH08b] only work for k-CNF formulas width $k \geq 4$, and in [BN08, BN11] we even have to choose $k \geq 6$ to find k-CNF formula families that optimally separate space and length and exhibit time-space trade-offs. We know from [ET01] that any 2-CNF formula is refutable in constant clause space, but should there not be 3-CNF formulas for which we could prove similar separations and trade-offs?

Given any CNF formula F, we can transform it to a 3-CNF formula by rewriting every clause $C = a_1 \vee \ldots \vee a_m$ in F with $m > 3$ as a conjunction of 3-clauses

$$\overline{y}_0 \wedge \bigwedge_{1 \leq i \leq m} (y_{i-1} \vee a_i \vee \overline{y}_i) \wedge y_n , \tag{14.12}$$

for some new auxiliary variables $y_0, y_1, \ldots, y_m$ unique for this clause C. Let us write $\widetilde{F}$ to denote the 3-CNF formula obtained from F in this way. It is easy to see that $\widetilde{F}$ is unsatisfiable if and only if F is unsatisfiable. Also, it is straightforward to verify that $L\big(\widetilde{F} \vdash 0\big) \leq L(F \vdash 0) + W(F) \cdot L(F)$ and $Sp\big(\widetilde{F} \vdash 0\big) \leq Sp(F \vdash 0) + \mathrm{O}(1)$. (Just note that each clause of F can be derived from $\widetilde{F}$ in length $W(F)$ and space $\mathrm{O}(1)$, and then use this together with an optimal refutation of F.)

It seems like a natural idea to rewrite pebbling contradictions $Peb_G[f]$ for suitable functions f as 3-CNF formulas $\widetilde{Peb}_G[f]$ and study length-space trade-offs for such formulas. For this to work, we would need *lower* bounds on the refutation clause space of $\widetilde{F}$ in terms of the refutation clause space of F, however.

Open Problem 26. *Is it true that $Sp\big(\widetilde{Peb}^2_G[\oplus] \vdash 0\big) \geq$ BW-Peb(G)? In general, can we prove lower bounds on $Sp\big(\widetilde{F} \vdash 0\big)$ in terms of $Sp(F \vdash 0)$, or are there counter-examples where the two measures differ asymptotically?*

This final open problem is of course of minor importance compared to the other, more fundamental questions considered in this section. However, we still find it interesting in the sense that if it could be shown to hold in general that $Sp\big(\widetilde{F} \vdash 0\big) \lessapprox Sp(F \vdash 0)$, then we would get all space lower bounds, and maybe also the length-space trade-offs, for free for 3-CNF formulas. It would be aesthetically satisfying not having to insist on using 6-CNF formulas to obtain these bounds. Incidentally, it would also further strengthen the argument that space should only be studied for formulas of fixed width (as was discussed above).

15 Some Open Questions

EDIT COMMENT 4: *The "raw material" for this concluding section is in place but the writing and editing remains to be done.*

We conclude this survey by briefly restating the open problems mentioned throughout this survey, with page references to where they are discussed in more detail.

15.1 Pebbling Problems

1. Can the exact black-white pebbling price be determined for pyramids of any height, even or odd (page 24)?

2. Is it possible to simplify the very intricate proof in [Kla85] of the pyramid pebbling price lower bound (page 37)?

3. Is it possible to find explicit or non-explicit superconcentrators with a smaller gap between the lower and upper bounds in Theorems 5.5 and 5.6? What about superconcentrator constructions that have appeared after [LT82] (page 41)?

4. Are there permutations for which the lower bound in Theorem 8.5 for pebblings of permutation graphs holds also for black-white pebbling (page 64)?

5. Is it possible to prove optimal time-space trade-offs on the form in Theorem 8.1 for any constant space (65)?

6. Is it possible to prove the black-white pebbling trade-offs in Section 9 for the original Carlson-Savage graphs yielding stronger trade-off parameters (page 69)?

7. Can the exponential trade-off curves in Theorems 6.2 and 11.2 be realized by a single family of graphs, or only for an infinite collection of families having threshold trade-offs for various points on the trade-off curves (page 79)?

8. Is it possible to prove a quadratic separation between black and black-white pebbling for polynomial-size DAGs (page 80)?

9. Can the problem of deciding the black-white pebbling price of graphs with bounded fan-in be shown to be PSPACE-complete (page 93)?

15.2 Proof Complexity Problems

10. Time-space trade-offs in practice? Open Problem 10 on page 111

11. PSPACE-completeness of total space in resolution? Open Problem 11 on page 112

12. PSPACE-completeness of clause space in resolution? Open Problem 12 on page 112

13. Combinatorial characterization of clause/formula space? Open Problem 13 on page 112

14. EXPTIME-completeness of resolution width? Open Problem 14 on page 112

15. Pebbling contradictions always refutable in black-white space? Open Problem 15 on page 117

16. Are bounded labelled pebblings as strong as general black-white pebblings? Open Problem 16 on page 117

17. Do OR-pebbling formulas inherit length-space trade-offs from underlying graphs? Open Problem 17 on page 122

18. Prove space lower bounds and time-space trade-offs for CP, PC and PCR via substitutions and projections? Open Problem 18 on page 125

19. Prove lower bounds on total space in resolution? Open Problem 19 on page 126

20. "Direct" proof that small space implies short length in resolution? Open Problem 20 on page 126

21. Stronger upper bounds on length in terms of clause space? Open Problem 21 on page 126

22. Is the length-width trade-off in Ben-Sasson and Wigderson necessary? Open Problem 22 on page 126
23. Are there polynomial length-width trade-offs for constant width? Open Problem 23 on page 127
24. Are there superpolynomial trade-offs for constant clause space? Open Problem 24 on page 127
25. Are there nontrivial trade-offs for linear or even superlinear clause space? Open Problem 25 on page 128
26. Do space lower bounds continue to hold if we convert a formula to 3-CNF? Open Problem 26 on page 128

Acknowledgements

I would like to thank Eli Ben-Sasson and Johan Håstad, with whom I have coauthored papers based on some of the material that appears in this survey, for many interesting and fruitful discussions. Also, I am grateful to David Carlson, John Gilbert, Nicholas Pippenger, and John Savage for helpful correspondence regarding their papers on pebbling, and to Joshua Buresh-Oppenheim, Nicola Galesi, Jan Johannsen, Pavel Pudlák, and Alexander Razborov for helping me to sort out some proof complexity related questions.

References

[AB04] Albert Atserias and Maria Luisa Bonet. On the automatizability of resolution and related propositional proof systems. *Information and Computation*, 189(2):182–201, March 2004. Preliminary version appeared in *CSL '02*.

[ABE02] Albert Atserias, Maria Luisa Bonet, and Juan Luis Esteban. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Information and Computation*, 176(2):136–152, August 2002. Preliminary version appeared in *ICALP '01*.

[ABLM08] Carlos Ansótegui, Maria Luisa Bonet, Jordi Levy, and Felip Manyà. Measuring the hardness of SAT instances. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI '08)*, pages 222–228, July 2008.

[ABMP01] Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is NP-hard to linearly approximate. *Journal of Symbolic Logic*, 66(1):171–191, March 2001.

[ABRW02] 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. Preliminary version appeared in *STOC '00*.

[ABRW04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM Journal on Computing*, 34(1):67–88, 2004.

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

[AD08] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, May 2008. Preliminary version appeared in *CCC '03*.

[AFT09] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. In *Proceedings of the 12th International Conference on Theory and Applications of Satisfiability Testing (SAT '09)*, volume 5584 of *Lecture Notes in Computer Science*, pages 114–127. Springer, 2009.

[AJPU07] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. *Theory of Computing*, 3(5):81–102, May 2007. Preliminary version appeared in *STOC '02*.

[AKV04] Albert Atserias, Phokion G. Kolaitis, and Moshe Y. Vardi. Constraint propagation as a proof system. In *Proceedings of the 10th International Conference on Principles and Practice of Constraint Programming (CP '04)*, volume 3258 of *Lecture Notes in Computer Science*, pages 77–91. Springer, 2004.

[Ale05] Michael Alekhnovich. Lower bounds for k-DNF resolution on random 3-CNFs. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, pages 251–256, May 2005.

[AR01] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '01)*, pages 190–199, October 2001.

[AT09] Albert Atserias and Marc Thurley. Personal communication, 2009.

[Ats04] Albert Atserias. On sufficient conditions for unsatisfiability of random formulas. *Journal of the ACM*, 51(2):281–311, March 2004.

[Ats06] Albert Atserias. Personal communication, 2006.

[BCIP02] Joshua Buresh-Oppenheim, Matthew Clegg, Russell Impagliazzo, and Toniann Pitassi. Homogenization and the polynomial calculus. *Computational Complexity*, 11(3-4):91–108, 2002. Preliminary version appeared in *ICALP '00*.

[Bea04] 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.

[BEGJ00] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, 2000. Preliminary version appeared in *FOCS '98*.

[Ben07] Eli Ben-Sasson. Personal communication, 2007.

[Ben09] Eli Ben-Sasson. Size space tradeoffs for resolution. *SIAM Journal on Computing*, 38(6):2511–2525, May 2009. Preliminary version appeared in *STOC '02*.

[BG01] Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, December 2001. Preliminary version appeared in *FOCS '99*.

[BG03] Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003. Preliminary version appeared in *CCC '01*.

[BGIP01] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *Journal of Computer and System Sciences*, 62(2):267–289, March 2001. Preliminary version appeared in *CCC '99*.

[BHJ08] Samuel R. Buss, Jan Hoffmann, and Jan Johannsen. Resolution trees with lemmas: Resolution refinements that characterize DLL-algorithms with clause learning. *Logical Methods in Computer Science*, 4(4:13), 2008.

[BI99] Eli Ben-Sasson and Russell Impagliazzo. Random CNF's are hard for the polynomial calculus. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS '99)*, pages 415–421, October 1999.

[BIPS10] Paul Beame, Russell Impagliazzo, Toniann Pitassi, and Nathan Segerlind. Formula caching in DPLL. *ACM Transactions on Computation Theory*, 1(3:9), March 2010. Preliminary version appeared in *CCC '03*.

[BIW04] Eli Ben-Sasson, Russell Impagliazzo, and Avi Wigderson. Near optimal separation of treelike and general resolution. *Combinatorica*, 24(4):585–603, September 2004.

[BJ10] Eli Ben-Sasson and Jan Johannsen. Lower bounds for width-restricted clause learning on small width formulas. In *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT '10)*, volume 6175 of *Lecture Notes in Computer Science*, pages 16–29. Springer, July 2010.

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

[BKS04] Paul Beame, Henry Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *Journal of Artificial Intelligence Research*, 22:319–351, December 2004.

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

[BN08] Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 709–718, October 2008.

[BN11] Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Proceedings of the 2nd Symposium on Innovations in Computer Science (ICS '11)*, January 2011. To appear.

[BP96] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science (FOCS '96)*, pages 274–282, October 1996.

[BP98] 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.

[BP07] Joshua Buresh-Oppenheim and Toniann Pitassi. The complexity of resolution refinements. *Journal of Symbolic Logic*, 72(4):1336–1352, December 2007. Preliminary version appeared in *LICS '03*.

[BPR95] Maria Bonet, Toniann Pitassi, and Ran Raz. Lower bounds for cutting planes proofs with small coefficients. In *Proceedings of the 27th Annual ACM Symposium on Theory of Computing (STOC '99)*, pages 575—584, May 1995.

[BW01] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001. Preliminary version appeared in *STOC '99*.

[CCT87] William Cook, Collette Rene Coullard, and Gyorgy Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, November 1987.

[CEI96] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 174–183, May 1996.

[Cha73] Ashok K. Chandra. Efficient compilation of linear recursive programs. In *Proceedings of the 14th Annual Symposium on Switching and Automata Theory (SWAT '73)*, pages 16–25, 1973.

[Chv73] Vašek Chvátal. Edmond polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, 4(1):305–337, 1973.

[CK02] Peter Clote and Evangelos Kranakis. *Boolean Functions and Computation Models*. Springer, 2002.

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

[CR79] Stephen A. Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979.

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

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

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

[CS88] Vašek Chvátal and Endre Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[DLL62] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem proving. *Communications of the ACM*, 5(7):394–397, July 1962.

[DP60] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7(3):201–215, 1960.

[EBL79] Peter van Emde Boas and Jan van Leeuwen. Move rules and trade-offs in the pebble game. In *Proceedings of the 4th GI Conference on Theoretical Computer Science*, volume 67 of *Lecture Notes in Computer Science*, pages 101–112. Springer, March 1979.

[EGM04] Juan Luis Esteban, Nicola Galesi, and Jochen Messner. On the complexity of resolution with bounded conjunctions. *Theoretical Computer Science*, 321(2-3):347–370, August 2004. Preliminary version appeared in *ICALP '02*.

[ET99] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. In *Proceedings of the 16th International Symposium on Theoretical Aspects of Computer Science (STACS '99)*, volume 1563 of *Lecture Notes in Computer Science*, pages 551–560. Springer, 1999.

[ET01] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Based on the conference papers in *STACS '99* [ET99] and *CSL '99* [Tor99].

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

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

[GL10] Nicola Galesi and Massimo Lauria. Optimality of size-degree trade-offs for polynomial calculus. *ACM Transactions on Computational Logic*, 2010. To appear.

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

[Gom63] Ralph E. Gomory. An algorithm for integer solutions of linear programs. In R.L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, New York, 1963.

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

[GT05] Nicola Galesi and Neil Thapen. Resolution and pebbling games. In *Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing (SAT '05)*, volume 3569 of *Lecture Notes in Computer Science*, pages 76–90. Springer, June 2005.

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

[HBPV08] Philipp Hertel, Fahiem Bacchus, Toniann Pitassi, and Allen Van Gelder. Clause learning can effectively P-simulate general propositional resolution. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI '08)*, pages 283–290, July 2008.

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

[HP07] Philipp Hertel and Toniann Pitassi. Exponential time/space speedups for resolution and the PSPACE-completeness of black-white pebbling. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*, pages 137–149, October 2007.

[HP10] Philipp Hertel and Toniann Pitassi. The PSPACE-completeness of black-white pebbling. *SIAM Journal on Computing*, 39(6):2622–2682, April 2010. Preliminary version appeared in *FOCS '07*.

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

[HU06] Alexander Hertel and Alasdair Urquhart. The resolution width problem is EXPTIME-complete. Technical Report TR06-133, Electronic Colloquium on Computational Complexity (ECCC), 2006.

[HU07] Alexander Hertel and Alasdair Urquhart. Game characterizations and the PSPACE-completeness of tree resolution space. In *Proceedings of the 21st International Workshop on Computer Science Logic (CSL '07)*, volume 4646 of *Lecture Notes in Computer Science*, pages 527–541. Springer, September 2007.

[HU09] Alexander Hertel and Alasdair Urquhart. Comments on ECCC report TR06-133: The resolution width problem is EXPTIME-complete. Technical Report TR09-003, Electronic Colloquium on Computational Complexity (ECCC), 2009.

[IPS99] Russell Impagliazzo, Pavel Pudlák, and Jiri Sgall. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[Iwa97] Kazuo Iwama. Complexity of finding short resolution proofs. In *Proceedings of the 22nd International Symposium on Mathematical Foundations of Computer Science 1997 (MFCS '97)*, volume 1295 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 1997.

[JN02] Jan Johannsen and N. S. Narayanaswamy. An optimal lower bound for resolution with 2-Conjunctions. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science (MFCS '02)*, volume 2420 of *Lecture Notes in Computer Science*, pages 387–398. Springer, August 2002.

[Joh83] David S. Johnson. The NP-completeness column: An ongoing guide. *Journal of Algorithms*, 4(4):397–411, December 1983.

[Joh01] Jan Johannsen. Exponential incomparability of tree-like and ordered resolution. Manuscript. Available at http://www.tcs.ifi.lmu.de/~jjohanns/notes.html, 2001.

[Kla85] Maria M. Klawe. A tight bound for black and white pebbles on the pyramid. *Journal of the ACM*, 32(1):218–228, January 1985. Preliminary version appeared in *FOCS '83*.

[Koz77] Dexter Kozen. Lower bounds for natural proof systems. In *Proceedings of the 18th Annual IEEE Symposium on Foundations of Computer Science (FOCS '77)*, pages 254–266, 1977.

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

[KS91] Balasubramanian Kalyanasundaram and George Schnitger. On the power of white pebbles. *Combinatorica*, 11(2):157–171, June 1991. Preliminary version appeared in *STOC '88*.

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

[Kul99] Oliver Kullmann. Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs. Technical Report TR99-041, Electronic Colloquium on Computational Complexity (ECCC), 1999.

[Len81] Thomas Lengauer. Black-white pebbles and graph separation. *Acta Informatica*, 16(4):465–475, December 1981.

[Lin78] Andrzej Lingas. A PSPACE-complete problem related to a pebble game. In *Proceedings of the 5th Colloquium on Automata, Languages and Programming (ICALP '78)*, pages 300–321, 1978.

[Lou79] Michael Conrad Loui. The space complexity of two pebble games on trees. Technical Memorandum MIT-LCS-TM-133, Laboratory for Computer Science, Massachusetts Institute of Technology, May 1979.

[Lou80] Michael Conrad Loui. A note on the pebble game. *Information Processing Letters*, 11(1):24–26, August 1980.

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

[LT82] Thomas Lengauer and Robert Endre Tarjan. Asymptotically tight bounds on time-space trade-offs in a pebble game. *Journal of the ACM*, 29(4):1087–1130, October 1982. Preliminary version appeared in *STOC '79*.

[LV83] G. Lev and Leslie G. Valiant. Size bounds for superconcentrators. *Theoretical Computer Science*, 22(3):233–251, February 1983.

[Mar08] João P. Marques-Silva. Practical applications of Boolean satisfiability. In *9th International Workshop on Discrete Event Systems (WODES '08)*, pages 74–80, May 2008.

[Mey79] Friedhelm Meyer auf der Heide. A comparison between two variations of a pebble game on graphs. Diplomarbeit, Universität Bielefeld, Fakultät für Mathematik, 1979.

[Mey81] Friedhelm Meyer auf der Heide. A comparison of two variations of a pebble game on graphs. *Theoretical Computer Science*, 13(3):315–322, 1981.

[NH08a] Jakob Nordström and Johan Håstad. Towards an optimal separation of space and length in resolution. Technical Report TR08-026, Electronic Colloquium on Computational Complexity (ECCC), February 2008.

[NH08b] Jakob Nordström and Johan Håstad. Towards an optimal separation of space and length in resolution (Extended abstract). In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, pages 701–710, May 2008.

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

[Nor09a] Jakob Nordström. Narrow proofs may be spacious: Separating space and width in resolution. *SIAM Journal on Computing*, 39(1):59–121, May 2009. Preliminary version appeared in *STOC '06*.

[Nor09b] Jakob Nordström. A simplified way of proving trade-off results for resolution. *Information Processing Letters*, 109(18):1030–1035, August 2009. Preliminary version in ECCC report TR07-114, 2007.

[Nor10a] Jakob Nordström. On the relative strength of pebbling and resolution (Extended abstract). In *Proceedings of the 25th Annual IEEE Conference on Computational Complexity (CCC '10)*, pages 151–162, June 2010.

[Nor10b] Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 2010. To appear. Available at the webpage http://www.csc.kth.se/~jakobn/research/.

[PH70] Michael S. Paterson and Carl E. Hewitt. Comparative schematology. In *Record of the Project MAC Conference on Concurrent Systems and Parallel Computation*, pages 119–127, 1970.

[Pin73] Mark S. Pinsker. On the complexity of a concentrator. In *Proceedings of the 7th International Teletraffic Conference*, pages 318/1–318/4, 1973.

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

[Pip78] Nicholas Pippenger. A time-space trade-off. *Journal of the ACM*, 25(3):509–515, July 1978.

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

[PT78] Wolfgang J. Paul and Robert Endre Tarjan. Time-space trade-offs in a pebble game. *Acta Informatica*, 10:111–115, 1978.

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

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

[PV76] Nicholas Pippenger and Leslie G. Valiant. Shifting graphs and their applications. *Journal of the ACM*, 23:423–432, July 1976.

[Raz98] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, December 1998.

[Raz03] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. Manuscript. Available at the webpage http://people.cs.uchicago.edu/~razborov/research.html, 2002-2003.

[Rei80] Rüdiger Reischuk. Improved bounds on the problem of time-space trade-off in the pebble game. *Journal of the ACM*, 27(4):839–849, October 1980.

[RM99] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, March 1999. Preliminary version appeared in *FOCS '97*.

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

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

[Sav98] John E. Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley, 1998. Available at the webpage http://www.modelsofcomputation.org.

[SBI04] Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM Journal on Computing*, 33(5):1171–1200, 2004. Preliminary version appeared in *FOCS '02*.

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

[Sch05] Uwe Schöning. Smaller superconcentrators of density 28. *Information Processing Letters*, 98(4):127–129, May 2005.

[Seg05] Nathan Segerlind. Exponential separation between Res(k) and Res($k+1$) for $k \leq \epsilon \log n$. *Information Processing Letters*, 93(4):185–190, February 2005.

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

[Set75] Ravi Sethi. Complete register allocation problems. *SIAM Journal on Computing*, 4(3):226–248, September 1975.

[SS77] Sowmitri Swamy and John E. Savage. Space-time trade-offs on the FFT-algorithm. Technical Report CS-31, Brown University, 1977.

[SS79] John E. Savage and Sowmitri Swamy. Space-time tradeoffs for oblivious interger multiplications. In *Proceedings of the 6th Colloquium on Automata, Languages and Programming (ICALP '79)*, pages 498–504, 1979.

[SS83] Sowmitri Swamy and John E. Savage. Space-time tradeoffs for linear recursion. *Mathematical Systems Theory*, 16(1):9–27, 1983.

[Tom78] Martin Tompa. Time-space tradeoffs for computing functions, using connectivity properties of their circuits. In *Proceedings of the 10th annual ACM symposium on Theory of computing (STOC '78)*, pages 196–204, 1978.

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

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

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

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

[Urq95] Alasdair Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 1(4):425–467, 1995.

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

[Van05] Allen Van Gelder. Pool resolution and its relation to regular resolution and DPLL with clause learning. In *Proceedings of the 12th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR '05)*, volume 3835 of *Lecture Notes in Computer Science*, pages 580–594. Springer, 2005.

[Wil88] Robert E. Wilber. White pebbles help. *Journal of Computer and System Sciences*, 36(2):108–124, 1988. Preliminary version appeared in *STOC '85*.

PROOF STRUCTURE COMMENT 1: *Last "proof comment". (This comment added to know how many there are in total.)*
EDIT COMMENT 5: *Last "edit comment". (This comment added to know how many there are in total.)*