Electronic Colloquium on Computational Complexity, Report No. 42 (2013)

Just a Pebble Game

Siu Man Chan*

UC Berkeley

March 24, 2013

Abstract

The two-player pebble game of Dymond–Tompa is identified as a barrier for existing techniques to save space or to speed up parallel algorithms for evaluation problems.

Many combinatorial lower bounds to study L versus NL and NC versus P under different restricted settings scale in the same way as the pebbling algorithm of Dymond–Tompa. These lower bounds include,

- the monotone separation of $\mathsf{m\text{-}L}$ from $\mathsf{m\text{-}NL}$ by studying the size of monotone switching networks in Potechin '10;
- a new semantic separation of NC from P and of NC^i from NC^{i+1} by studying circuit depth, based on the techniques developed for the semantic separation of NC^1 from NC^2 by the universal composition relation in Edmonds–Impagliazzo–Rudich–Sgall '01 and in Håstad–Wigderson '97; and
- the monotone separation of $\mathsf{m\text{-}NC}$ from $\mathsf{m\text{-}P}$ and of $\mathsf{m\text{-}NC}^i$ from $\mathsf{m\text{-}NC}^{i+1}$ by studying
  - the depth of monotone circuits in Raz–McKenzie '99; and
  - the size of monotone switching networks in Chan–Potechin '12.

This supports the attempt to separate NC from P by focusing on depth complexity, and suggests the study of combinatorial invariants shaped by pebbling for proving lower bounds. An application to proof complexity gives tight bounds for the size and the depth of some refinements of resolution refutations.

1 Introduction

Memory space and parallel time are two important resources of deterministic computation. To study these two resources, researchers considered different approaches. This paper focuses on the approach of analyzing pebble games and the approach of analyzing concrete combinatorial models of computation.

It turns out that there is an unobserved connection between the two approaches. Namely, many of the combinatorial approaches for studying L versus NL and NC versus P under different restricted settings implicitly proved a lower bound scaling in the same way as the pebbling algorithms. This combinatorial coincidence for different analyses under different restrictions calls for further studies.

*siuman@cs.berkeley.edu. This material is based upon work supported by the National Science Foundation under Grant No. CCF-1017403 and Grant No. CCF-0830797.

ISSN 1433-8092

1.1 Pebble Games

Pebble games were introduced for studying programming languages and compiler construction. The dependency in data flow is modeled by a directed acyclic graph of bounded in-degree, and the pebble games emulate the register allocation and resource usage in the flow of data over the graph. As another closely related example in database systems, a directed acyclic graph models the referential structure of tables in a database,[1] and the pebble games emulate the data access pattern executed by a certain query.

In terms of computational resources, deterministic space is traditionally emulated by *the number of (black) pebbles* required in a one-player pebble game [PH70, Set75], and parallel time (or more accurately, alternating time) is traditionally emulated by the *time* required in a two-player pebble game introduced by Dymond and Tompa [DT85].

Upper Bounds The study of these pebble games led to non-trivial algorithms, upper bounding resource requirements. For space, Hopcroft, Paul, and Valiant [HPV77] showed that any graph of bounded in-degree on t vertices requires at most $O(t/\log t)$ pebbles in the one-player game, implying that a time t (deterministic) computation requires at most $O(t/\log t)$ space, i. e., $\mathsf{DTime}[t] \subseteq \mathsf{DSpace}[t/\log t]$.

For parallel time, Dymond and Tompa [DT85] showed that any graph of bounded in-degree on t vertices requires at most $O(t/\log t)$ time in the two-player game, strengthening the above result to imply that a time t (deterministic) computation requires at most $O(t/\log t)$ alternating time,[2] i. e., $\mathsf{DTime}[t] \subseteq \mathsf{ATime}[t/\log t]$. Alternating time measures the time spent on an alternating machine [CKS81], which is a natural model of (deterministic) parallel computation, and hence the result of Dymond and Tompa implied speedups of parallel time (when the number of processors is unbounded).

Pebble Games and Complexity Classes Also, certain relationships among different resources of computation can be recast as pebbling results. For example, (a slight variant of) the two-player pebble game of Dymond and Tompa [VT89] (1) exactly *characterizes* the parallelism of different complexity classes (e. g., NC and P); and (2) can re-derive known complexity results, including the simulation of Savitch [Sav70] showing that $\mathsf{NL} \subseteq \mathsf{DSpace}[\log^2 n]$.

Lower Bounds and Trade-Offs The study of pebble games also gave lower bounds on resource requirements or indicating trade-offs of different resources *in restricted models of computation.*

Paul, Tarjan, and Celoni [PTC76] constructed a graph of bounded in-degree on t vertices which requires $\Omega(t/\log t)$ pebbles in the one-player game emulating space; and by a simulation argument in pebble games, this graph also requires $\Omega(t/\log t)$ time in the two-player game emulating alternating time [DT85]. These lower bounds are tight given the upper bounds on pebble games. To the best of our knowledge, we still don't know how to save more space or alternating time (a measure of parallel time) than the pebbling algorithms for a P-complete problem.[3]

In addition to the **black pebble game** and the **Dymond–Tompa pebble game**, two other pebble games were used in the combinatorial approach for proving restricted lower bounds (to be discussed in § 1.3).

Raz–McKenzie pebble game Raz and McKenzie [RM99] introduced a two-player pebble game over a directed acyclic graph, motivated by the depth complexity of decision trees solving

[1] In reality, the referential structure can have cycles and have large in-degree. We ignore such complications in this exposition.

[2] The result on alternating time is stronger, since $\mathsf{ATime}[t] \subseteq \mathsf{DSpace}[t]$ [CKS81].

[3] For example, concerning circuit depth (to be introduced next), although there are some non-pebbling algorithms for trading circuit depth for (semi-unboundedness of) fan-in [LV03, Wil05], those algorithms do not give a saving in depth when simulated on circuits of *bounded fan-in.*

search problems [LNNW95]. The pebble game was first used for proving lower bounds on monotone alternating time (see § 1.3). Later, it was applied to proof complexity, e. g., [BEGJ98], and inspired the use of *pebbling contradictions* which form the basis of most time-space trade-offs and many separation results in proof complexity (to be discussed in § 1.5). Elias and McKenzie [EM10] made explicit the role of the pebble game in the monotone results, and initiated a study of the pebble cost over different directed acyclic graphs.

Reversible pebble game Bennett [Ben73] initiated the study of reversible computation as a possibility to eliminate (or significantly reduce) energy dissipation in logical computation. Reversible computation is increasingly important (i) because computing chips are getting smaller and energy dissipation is becoming an issue; and (ii) because observation-free quantum computation is inherently reversible. Bennett studied the time and space complexity in reversible simulation of irreversible computation, and as an abstraction mentioned *reversible pebble game* [Ben89], which is the reversible version of the black pebble game. This led to the study of the reversible pebble game over different directed acyclic graphs [LV96, Krá01b] and its relation to time-space trade-offs in reversible simulation of irreversible computation [LTV97, LMT00, Wil00, BTV01]. Later, in the combinatorial approach, Potechin [Pot10] independently and implicitly used the reversible pebble game (made explicit in [CP12]) for proving lower bounds on monotone space complexity (see § 1.3).

The reader is referred to the literature for further discussions on the black pebble game [Nor12, Krá01b], the Dymond-Tompa pebble game [DT85, VT89], the Raz–McKenzie pebble game [RM99, EM10], and the reversible pebble game [Ben89, Krá01b].

1.2 Our Results in Pebble Games

Theorem 1 (Just a Pebble Game)**.** *The Dymond–Tompa pebble game, the Raz–McKenzie pebble game, and the reversible pebble game of Bennett have the same pebble cost. That is, for any directed acyclic graph having a unique sink vertex, the following are equivalent for pebbling the sink vertex:*

- *it takes h time in the Dymond–Tompa pebble game (§ 3.1);*
- *it takes h time in the Raz–McKenzie pebble game (§ 3.2);*[4] *and*
- *it takes h pebbles in the reversible pebble game of Bennett (§ 3.3).*[5]

Corollary 1.1 (Upper Bounds on Pebble Costs)**.** *Any directed acyclic graph on n vertices with bounded in-degree has cost at most $O(n/\log n)$ in the Raz–McKenzie pebble game or the reversible pebble game.*

Corollary 1.2 (Raz–McKenzie versus Black Pebbling)**.** *The Raz–McKenzie pebble cost is at least the (irreversible) black pebble cost.*[5]

Remark 1.3 (Connections in Pebble Games). Theorem 1 establishes a connection among different pebble games introduced for very different reasons.

1. It strengthens and explains the simulation result of Dymond–Tompa [DT85], which states that the Dymond–Tompa pebble cost of a graph is at least the black pebble cost of a graph, mirroring the inclusion $\mathsf{ATime}[t] \subseteq \mathsf{DSpace}[t]$. It is because the reversible pebble cost is at least the black pebble cost.

[4] This solves an open problem raised in Elias–McKenzie [EM10] for connecting their pebble game with other pebble games.

[5] In the reversible pebble game, it is required to pebble the sink vertex and to remove pebbles on all other vertices.

2. It explains some of the known results in computational complexity to be reviewed next (§ 1.3).

3. It explains some of the known results in proof complexity to be reviewed next. In particular, Corollary 1.2 gives a new connection between two pebble games studied in proof complexity (see § 1.5).

4. It connects the pebbling results in the Dymond–Tompa pebble game [DT85], the Raz–McKenzie pebble game [RM99, EM10], and the reversible pebble game [LV96, Krá01b, Pot10, CP12] over different directed acyclic graphs, e. g., line graphs, pyramid graphs, butterfly graphs, or the hard-to-pebble graph in [PTC76]. For example, this transfers the tight bound of $\Theta(n/\log n)$ pebbles for the graph in [PTC76] over the Dymond–Tompa pebble game to the Raz–McKenzie pebble game and to the reversible pebble game, which was not known before.

To better understand the connection between pebble games and complexity, we next review the combinatorial approach for proving lower bounds.

1.3 Combinatorial Models of Computation

We briefly recall two combinatorial models of computation which characterize parallel time and memory space. We ignore the issues of uniformity in this paper, by assuming that the combinatorial models are sufficiently uniform. Alternatively, the reader may want to append /poly to every machine-based complexity class.

Circuits Parallel time is modeled by the *depth* of a circuit: $\mathsf{ATime}[t] = \mathsf{Depth}[t]$ [Ruz81]. Recall that $\mathsf{ATime}[\cdot]$ refers to alternating time, a measure of parallel time on alternating machines. This paper considers *boolean* circuits *of bounded fan-in* unless otherwise noted.

Switching Networks Memory space of a deterministic computation is modeled by the *size* of a switching network: $\mathsf{DSpace}[s] = \mathsf{SNSize}[2^{\Theta(s)}]$. A switching network computes by reachability in a symmetric way, where the symmetry/reversibility mirrors determinism [LMT00, Rei08]. The direction $\mathsf{DSpace}[s] \subseteq \mathsf{SNSize}[2^{\Theta(s)}]$ is folklore [Lee59] (e. g., see [Pot10, §2]), and $\mathsf{SNSize}[2^{\Theta(s)}] \subseteq \mathsf{DSpace}[s]$ is proved by Reingold [Rei08].

Researchers commonly add restrictions to the combinatorial models to get lower bounds in restricted settings. We recall two such restrictions below.

Monotone Restriction A boolean function is monotone if flipping an input bit from False to True cannot flip the output bit from True to False. When computing monotone boolean functions, it is common to add a monotone restriction to the model, which is to disallow logical negation. Monotone restriction applies to circuits and switching networks naturally (as opposed to e. g., Turing machines).

Problem-Specific Restriction In addition to the syntactic restriction of monotonicity, researchers also studied different semantic restrictions. Sometimes, the semantic restriction is designed with the computational problem in mind. We give one example below.

Karchmer and Wigderson [KW90] characterized the depth complexity of circuits as the communication complexity of a two-party game. To explore the intuition given by the communication game, and in particular whether depth complexity scales with iterated composition of hard functions (i. e., direct-sum phenomenon), Karchmer, Raz, and Wigderson [KRW95, §6] invented a communication game called *universal composition relation* to model iterated composition of hard functions, where the structure of iterated composition forms a tree (of branching factor and height about $\log n$). Roughly, the universal composition relation is similar to a standard communication game, except that the parties are required to output a bit on some leaf node of the tree (intuitively, the parties need to locate a branch of the tree leading to the leaf node, hence the communication complexity should scale with the height of the tree). Note that this restriction makes sense only for the problem of universal composition relation (unlike the monotone restriction, which applies to any monotone boolean function), and also only in the model of communication game.

Previous Results

For the depth complexity of semantically restricted circuits, Edmonds, Impagliazzo, Rudich, and Sgall [EIRS01] employed an information-theoretic counting argument to show that the universal composition relation of d levels of k-bit boolean function requires $dk - O\big(d^2(2\log k)^{1/2}\big)$ bits of communication when $d = \log n/\log\log n$ and $k = \log n$. Håstad and Wigderson [HW97] subsequently constructed a sub-additive measure to show that the universal composition relation of d levels of k-bit boolean function requires $\big(1 - o(1)\big)dk$ bits of communication when $d = o(\sqrt{k/\log k})$ and $k = \log n$. Both results suggest the *semantic analogue* of the separation $\mathsf{NC}^1 \subset \mathsf{NC}^2$, where NC^i are circuits of polynomial size and of $O\big(\log^i(n)\big)$ depth.

For the depth complexity of monotone circuits, Karchmer and Wigderson [KW90] introduced the communication game framework to prove that the NL-complete problem of directed connectivity requires $\Omega(\log^2 n)$ depth on monotone circuits,[6] implying $\mathsf{m\text{-}NC}^1 \subset \mathsf{m\text{-}NL} \subseteq \mathsf{m\text{-}NC}^2$. Raz and McKenzie [RM99] extended the information-theoretic argument of Edmonds–Impagliazzo–Rudich–Sgall to reprove the directed connectivity result of Karchmer–Wigderson,[6] and in addition showed the separation of $\mathsf{m\text{-}NC} \subset \mathsf{m\text{-}P}$ and of $\mathsf{m\text{-}NC}^i \subset \mathsf{m\text{-}NC}^{i+1}$, by studying the P-complete problem of Generation. Subproblems of Generation with additional restrictions on the structure of 'the generation graph' are complete for smaller complexity classes like non-deterministic logspace (NL) and Nick's class (NC) [JL74, BM91], where the generation graph refers to the structure of certificate in the Yes-instances (see e. g., [CP12, § 3.2]).[7]

The results on monotone circuits were subsequently strengthened to monotone switching networks, i. e., from (monotone) alternating time to (monotone) deterministic space.[8] Departing from the communication game of Karchmer–Wigderson which forms the basis of most results concerning circuit depth [BS90, KW90, GH92, RW92, KRW95, GS95, RM99, EIRS01, Joh01], Potechin [Pot10] introduced a Fourier analytic framework, proving that directed connectivity requires monotone switching networks of size $n^{\Omega(\log n)}$, which can be interpreted as proving $\mathsf{m\text{-}L} \subset \mathsf{m\text{-}NL}$ on monotone switching networks.[9] The Fourier analytic framework is recently reinterpreted as describing an enumerative-combinatorial invariant [CP12], and the lower bound of Raz–McKenzie on Generation is strengthened to monotone switching networks. For further discussion on the switching network model or the Generation problem, see the references in [CP12].

As mentioned in § 1.1, two pebble games were used in the monotone results. In general, the monotone circuit depth for Generation scales as $\Omega(h\log n)$ when $h \leq n^{O(1)}$ is the Raz–McKenzie pebble cost of the generation graph [EM10]; and the monotone switching network size for the

[6] Karchmer–Wigderson [KW90] and Raz–McKenzie [RM99] in fact proved the same lower bound of $\Omega(\log^2 n)$ for the depth of monotone circuits solving *un*directed connectivity, which is L-complete [Rei08].

[7] We focus on the depth complexity of efficient problems, i. e., inside P or $\mathsf{m\text{-}P}$ under suitable restrictions, and did not mention, e. g., the lower bounds of k-clique [Raz85, AB87, GH92, Hak95, RM99, CP12] or matching [RW92] on monotone circuits.

[8] By a simulation argument mirroring $\mathsf{ATime}[t] \subseteq \mathsf{DSpace}[t]$ (see e. g., [CP12, §1]), a lower bound of $2^{\Omega(t)}$ on the size of (monotone) switching networks translates to a lower bound of $\Omega(t)$ on the depth of (monotone) circuits, hence the result on monotone switching network is stronger.

[9] It should be noted that there are at least two combinatorial models for (non-uniform) $\mathsf{m\text{-}L}$ in the literature: as monotone (boolean) circuits (of bounded fan-in) of logarithmic *width* and polynomial size [GS95, Gri91], or as monotone switching networks of polynomial size [Raz91, Pot10]. It appears that the two models are not comparable. This work focuses on monotone switching networks of polynomial size as the combinatorial model for (non-uniform) $\mathsf{m\text{-}L}$.

Generation problem scales as $n^{\Omega(h)}$ when $h \le n^{O(1)}$ is the reversible pebble cost of the generation graph [CP12].[10]

1.4 Our Results in Computational Complexity

Theorem 1 has the following consequence by the discussion in § 1.3

Corollary 1.4 (Improved Bounds for Generation)**.** *For any directed acyclic graph G, for the sub-problem of Generation where the generation graph is restricted to G, the lower bound on the size of monotone switching networks [CP12] implies the lower bound on the depth of monotone circuits [EM10] up to constant factors.*

In addition, at a high level, we combine the semantic separation of circuit depth [HW97, EIRS01] with the framework of Dymond–Tompa game. Instead of considering the *iterated multiplexor* problem of universal composition relation with a tree structure [KRW95, §6], we consider the *iterated indexing* problem over any directed acyclic graph. This minor twist completely changed the combinatorics of the problems. Our computational problem, called DAG evaluation (Definition 4.1), is a generalization of the tree evaluation problem considered by Cook, McKenzie, Wehr, Braverman, and Santhanam [CMW+12].[11] The DAG evaluation problem is a slight variant of the P-complete problem of circuit evaluation, and it captures the combinatorial essence of the Generation problem discussed above [McK10, EM10].

For this computational problem, we consider a problem-specific restriction called *output-relevant* circuits (§ 4.2). Roughly, in terms of the two-party communication game of Karchmer and Wigderson [KW90], a circuit is output-relevant if the two parties are required to output a relevant bit, which is a more natural restriction (than the output-leaf restriction in the universal composition relation) for studying depth complexity.[12] In particular, it is unclear how to turn the universal composition relation into a proper Karchmer–Wigderson game (so that it corresponds properly to circuit depth), which is *not* the case for output-relevant circuits.

Theorem 2 (Pebbling is Optimal)**.** *Consider a directed acyclic graph G whose Dymond–Tompa game takes h time. Any output-relevant circuit solving the DAG evaluation problem over G of bit-length k has depth $\Omega(hk)$ when $2^k \ge |V|^{\Theta(1)}$.*

Theorem 2 is complemented by a matching upper bound, that there is a circuit of depth $O(hk)$ implementing the pebbling algorithm of Dymond–Tompa. Unlike previous bounds on monotone circuits [RM99, Joh01, EM10] which are tight up to $n^{\Theta(1)}$, our bounds on restricted circuits are tight up to multiplicative factors. The tight bound can be interpreted as the semantic separation of NC from P and of NC^i from NC^{i+1}, by considering the pyramid graph of height $\Theta(\log^i n)$. In terms of circuit depth,[13] Theorem 2 gives an exponential improvement on an incomparable (but more natural[12]) model over previous results [EIRS01, HW97], which suggested only a semantic separation of NC^1 from NC^2.

[10] The proof in the journal vesrion of [CP12] clearly works for any directed acyclic graph.

[11] The generalization to DAG is also considered by Wehr [Weh11]. For comparison, Wehr studied the branching program model, and proved a lower bound (instead of a tight bound) in terms of the black pebble cost of the directed acyclic graph under a relatively restricted setting.

[12] Output-relevance is motivated by the efficiency of shallowly packing certificates for use by two competing provers, the alternation of which governs the combinatorial recurrence behind both the Dymond–Tompa game and the Karchmer–Wigderson game. For further justification, see Remark 4.9.

[13] We will discuss some related approaches for separating complexity classes in § 6, including approaches that consider both the size and depth of a circuit, e. g., by algebro-geometric invariants [Mul99, MS01, MS08], multi-party communication complexity [Cha07, BC08], competing-prover protocols [KR13], and block-respecting simulations [LW12].

Remark 1.5 (Circuit Depth and NC versus P). Theorem 2 supports the attempt to separate NC from P (and NC^i from NC^{i+1}) by focusing on circuit *depth*. By connecting (non-monotone but restricted) circuit depth with the Dymond–Tompa game, Theorem 2 gives evidence to support the attempt to study circuit depth *alone* (as opposed to a combination of depth and size) for separating NC from P,[13] due to very similar recurrence in the minimization of the depth complexity in the Karchmer–Wigderson game and in the Dymond–Tompa game.[14] More importantly, now Theorems 1 and 2 together put many of the existing combinatorial lower bounds concerning circuit depth for separating NC from P [KRW95, EIRS01, HW97, RM99, Joh01, Pot10, CP12] into the Dymond–Tompa game framework. This connection explains the same scaling in lower bounds given by apparently different pebble games: there is just one pebble game in disguise.[15] However, this raises an interesting follow-up question: why do the different analyses for different combinatorial arguments under different restricted settings converge to the same pebble game (which basically characterizes parallelism)? Also, the Dymond–Tompa game lower bounds the depth complexity on these restricted computational models, so how far (i. e., how general a model) does this correspondence hold?

We next briefly review the motivation for studying proof complexity, before we state our results on the depth of resolution refutations.

1.5 Proof Complexity

The study of proof complexity was initiated by Cook and Reckhow [CR79], who showed that NP = co-NP iff there is an efficient (i. e., polynomially bounded) proof system. Since its introduction, proof complexity has been studied by many researchers. We mention below two such motivations relevant to this paper.

Combinatorial methods for studying computational complexity One way to approach the distant goal of separating P from NP is to show that NP $\neq$ co-NP (since P = co-P), by proving super-polynomial lower bounds on successively stronger proof systems for propositional tautologies. Hence proving combinatorial lower bounds on proof systems can be seen as sharpening our combinatorial tools for eventually separating complexity classes, if possible.

Analysis of practical automated theorem-provers Lower bounds and trade-off results for seemingly weak and restricted proof systems already apply to the performance characteristics of most of the automated theorem-provers used in practice. For example, after failing to search for a satisfying assignment, the execution of the proof search algorithm in [DP60, DLL62] (known as DLL or DPLL) corresponds to a refinement (i. e., a restricted version) of resolution refutation whose structure forms a tree, hence called a *tree-like* resolution. Resolution refutation is a weak proof system in theory but widely used in practice. The study of trade-off results, or the comparison of different variants of proof systems (e. g., tree-like versus general), have consequences to the performance of different proof search strategies used in practice (see, e. g., [JMNŽ12]).

[14] Another evidence was the monotone separation of m-NC from m-P (and of $\mathsf{m\text{-}NC}^i$ from $\mathsf{m\text{-}NC}^{i+1}$) by Raz–McKenzie [RM99], where the lower bound on depth holds regardless of size (also implied by its strengthening [CP12]), although this monotone evidence is weak due to known exponential separations of monotone depth from non-monotone depth, e. g., for matching [RW92].

[15] For example, this may explain why in the Fourier analytic framework [Pot10], it is sufficient to consider reversible pebbling configurations [CP12] instead of knowledge sets [Pot10]. Also, Corollary 1.4 completes the picture of simulation results between circuits and switching networks, for the subproblem of Generation whose generation graph is *any directed acyclic graph* (in addition to the line graphs or the pyramid graphs known previously).

Resources: Size, Space, and Rank

Out of the many resources considered for studying proof complexity, we mention below three resources relevant to this paper.

Size The *size* of a refutation is the number of clauses,[16] or equivalently (up to a factor of two) the number of derivation steps. Hence the size complexity lower bounds the running time of a certain class of proof search algorithms (even allowing non-determinism). The size complexity is widely regarded as the most important complexity measure.

Space Among others, the *space* of a refutation may count the number of clauses (*clause space*) or the number of variables (*variable space*[17]) in any configuration in a refutation. Hence the space complexity measures the memory requirement (which is often a limiting resource for clause learning) of a certain class of proof search algorithms. Space complexity (in the configuration-style) was introduced by Esteban and Torán [ET01] and extended by Alekhnovich, Ben-Sasson, Razborov, and Wigderson [ABSRW02].

Rank The *rank* of a refutation measures the sequentiality of a certain class of proof search algorithms, e. g., for resolution-based proof systems, it is *depth*; and for semi-algebraic proof systems (i. e., polynomial threshold proof systems like Gomory–Chvátal cutting planes or Lasserre/Positivstellensatz), it is the number of *rounds*. At a high level, the rank of many proof systems may be related: the rank (depth) of the weak proof system of resolution is related to another measure called *width* [Urq11, Ber12], which in certain cases can be used for proving a rank (round) lower bound on the very strong proof system of Lasserre/Positivstellensatz [Gri01,Sch08,Tul09,Cha12].[18] The depth of resolution refutations was first systematically studied by Urquhart [Urq11], and the number of rounds of different semi-algebraic proof systems have been routinely studied, e. g., in proof complexity [BOCIP02, BOGH+06] or in hardness of approximation [Sch08, Tul09, Cha12].

There are some known relationships among different resources, connecting the most important resource of size to other resources. This gives another justification for studying space and rank.

Space and Size Clause space upper bounds (with some loss and via another measure *width*) the logarithm of size for resolution [AD08]. As a partial converse, the logarithm of size upper bounds clause space for tree-like resolution [ET01]. As for variable space, a lower bound on variable space can be escalated to a lower bound on clause space via substitution [BSN11], and this connection yielded one of the tightest size-space trade-offs currently known in proof complexity by studying pebbling contradictions [BSN11].

Rank and Size Urquhart argued that rank is significant since "all proofs of resolution size lower bounds implicitly prove depth lower bounds" [Urq11]. In practice, there are natural rank-based procedures for generating refutations in some proof systems, e. g., the Davis–Putnam procedure for resolution [DP60], (a variation of) the Gröbner basis algorithm for Polynomial Calculus [CEI96], and the semi-definite programming of Lasserre/Positivstellensatz [Gri01, Las01]. In this sense, rank measures the time needed for *deterministically* generating refutations in many practical proof systems, and for them rank may be as important as size (e. g., in [Gri01, Las01, Sch08, Tul09, Cha12]).

[16] Some literature calls this measure *length*, reserving size as the total number of symbols in a refutation (see e. g., the survey by Nordström [Nor12]). The two measures are polynomially related, and are used interchangeably in this paper.

[17] The term *variable space* was used in the literature to mean two related but different concepts: the number of literals counted with repetitions, or the number of variables counted without repetitions. The latter meaning, which is recently becoming the standard usage [Nor12, Footnote 5], is used here.

[18] A paper even suggests that any rank lower bound on resolution can be directely translated (with some loss) into a rank lower bound on some strong proof systems including Lasserre [BHP10], but an anonymous reviewer claims that this proof is broken.

Previous Results

The pebbling approach is routinely studied in proof complexity, in the form of *pebbling contradictions*, i. e., an unsatisfiable formula with one boolean variable per vertex, stating that (1) all source variables are true; (2) truth propagates through the graph; and (3) some sink variable is false. Often, certain pebbling properties (e. g., time and space) of the underlying graph is escalated to the formula via substitution [BSN11] or lifting [HN12].

The study of pebbling contradictions gave many of the best known separations (of different proof systems) and trade-offs (of different resources). In particular, the **Raz–McKenzie pebble game** has been used for separating tree-like and general cutting plane refutations [BEGJ98], and the (irreversible) **black pebble game** has been used for separating tree-like and general resolution size [BSIW04, Urq11], regular and general resolution size [AJPU07], DPLL (tree-like resolution) and a theoretical proof system based on clause learning algorithms [BIPS10], Nullstellensatz and Polynomial Calculus degree [BOCIP02], and the hierarchy of tree-like k-DNF-resolution and general resolution size [EGM04].[19]

1.6 Our Results in Proof Complexity

Let Σ_G denote the pebbling contradiction over G (Definition 5.2, see also [Nor12, Urq11]). The substitution construction of Alekhnovich–Razborov [BS09] is denoted $\Sigma^{\oplus}$ below; for generalizations, see [BSN11]. Denote $\textsc{Val}(G)$ as the value of the graph G, i. e., the pebble cost in the Dymond–Tompa game, or equivalently, in the Raz–McKenzie pebble game or in the reversible pebble game (Theorem 1).

Theorem 3 (Depth of Pebbling Contradictions)**.** *Fix a directed acyclic graph $G = (V, E)$ with a unique sink τ. The depth complexity of resolution refutation for Σ_G is* exactly *the pebble cost in the Raz–McKenzie pebble game to pebble the sink vertex of $\hat{G}$, where $\hat{G} := (V \cup \{\hat{\tau}\}, E \cup \{(\tau, \hat{\tau})\})$ is G augmented with an extra vertex $\hat{\tau}$ as the new sink.*

It is easy to see that the variable space needed for resolution refutation of Σ_G is at most the (irreversible) black pebble cost of G by simulating a black pebbling strategy. Take G to be the line graph on n vertices, this gives a separation of variable space (at most 2) and depth (at least $\log n$), solving an open problem raised by Urquhart [Urq11, Problem 7.2].

Theorem 4 (Tight Size Bounds for Tree-Like Resolution)**.** *The tree-like resolution refutation of $\Sigma_G^{\oplus}$ has size complexity $2^{\Theta(\textsc{Val}(G))}$.*

Remark 1.6 (Decision Tree and Reversible Pebble Game)*.* Theorem 3 gives an exact characterization, improving on the lower bound of Urquhart [Urq11]. Exact combinatorial characterization can be useful for translating results to different settings, e. g., Berkholz [Ber12] recently connected the exact combinatorial characterization of resolution width [AD08] with the combinatorial game of Kasai–Adachi–Iwata [KAI79, AIK84], proving an unconditional time lower bound. Theorem 4 can be seen as a result in this direction.

Moreover, this shows that the lower bounds in previous works [BSIW04, BIPS10, EGM04], in particular those concerning the depth of resolution refutations [Urq11], the degree of Polynomial Calculus [BOCIP02][20], and the size of tree-like cutting plane refutations [BEGJ98], morally follow

[19] We did not mention the use of black-white pebbling for time-space trade-offs [BS09, BSN11], see e. g., [Nor12].

[20] Buresh-Oppenheim, Clegg, Impagliazzo, and Pitassi only claimed the result in terms of the (irreversible) black pebble game, but it appears that their proof [BOCIP02, Lemma 4.10] works also in terms of the reversible pebble game, due to its combinatorial recurrence (Proposition 3.18 and Corollary 3.26).

from the pebble cost of a single pebble game, wearing different costumes listed in Theorem 1. Since the Dymond–Tompa game and the Raz–McKenzie pebble game were introduced for studying depth complexity, this may explain the use of the (reversible) black pebble game in Theorem 3 and in previous works.

Recall that k-DNF-resolution (Definition 5.3) extends the usual resolution.

Theorem 5. *Any k-DNF-resolution refutation of Σ_G has depth at least $1 + (\textsc{Val}(G) - 1)/k$.*

It is not hard to see that the lower bound should worsen with k, the arity of the DNF resolution. For constant k (which roughly corresponds to the case of boolean circuits of bounded fan-in), this lower bound is tight up to constant factors.

1.7 Techniques

The equivalence of the pebble games is proved by simulation arguments, on observing their similarities in combinatorial recurrence. The results on restricted models fall into three categories: (1) circuits under semantic restriction (thrifty circuits versus output-relevant circuits); (2) computational models under monotone restriction (monotone circuits versus monotone switching networks); and (3) weak proof systems (resolution refutations versus k-DNF-resolution refutations). Note that in all cases, the second model simulates (thus is stronger than) the first model.

All the upper bounds proved in this work are given by a pebbling strategy (of one of the pebble games listed in Theorem 1),[21] implemented in the weaker models. As for the lower bounds in slightly stronger models, although the computation appears not to follow a pebbling strategy, morally we can always decode an underlying strategy (or a family of strategies). In other words, the hardness of pebbling is escalated to the hardness in the respective, slightly stronger models.

As for the actual execution of the lower bound arguments, we consider the specifics of the models: (1) for circuits under semantic restriction, our lower bound is based on the extension by Raz–McKenzie [RM99] of the information-theoretic adversary argument by Edmonds–Impagliazzo–Rudich–Sgall [EIRS01]; (2) for computational models under monotone restriction, the lower bounds are based on the extension by Chan–Potechin [CP12] of the framework of invariants by Potechin [Pot10] or the extension by Raz–McKenzie [RM99] mentioned above (first proved in [EM10], see Theorem 13); and (3) for weak proof systems, our lower bound is an adversary argument based on the recurrence of the Raz–McKenzie pebble game.

1.8 Organization

Preliminary definitions and conventions are collected in § 2.

The three pebble games are introduced, and proved equivalent, in § 3. The equivalence of the three pebble games (Theorem 1) follows from Theorems 6 and 7.

The DAG evaluation problem is treated in § 4, which proves the lower bound of Theorem 2 as Theorem 10, based on the information theoretic counting arguments in Appendix A. The lower bound is complemented by a matching upper bound, proved as Theorem 9. Proposition 4.10 shows that output-relevant circuits simulate thrifty circuits.

The complexity of resolution refutations is studied in § 5, which proves Theorems 3 to 5.

Other approaches for separating complexity classes around P are discussed in § 6, and future directions are listed in § 7.

[21] Note that the upper bound for the problem of Generation on monotone models is not given by an optimal pebbling strategy, unlike other problems considered here, e. g., graph reachability and the DAG evaluation problem.

The Appendix B studies the nondeterministic version of the computational problem, proving Theorem 13 via the Dart game framework of Raz–McKenzie [RM99].

2 Preliminaries

Denote $[n] := \{0, 1, \ldots, n-1\}$. A subset S of a set A is identified with its indicator function $\chi_S \in 2^A \cong \{0,1\}^A$, where $\chi_S(i) = 1$ iff $i \in S$.

Notation 2.1 (Restriction)**.** The notation $\restriction$ will be overloaded for different (non-conflicting) definitions. In general, for a tuple x in a product space $X := A^B$ where A and B are sets, $x\restriction_b := x_b \in A$ denotes the entry of x indexed by $b \in B$. However, there are two exceptions: (1) for instances to the evaluation problem BDEP_G^k (Notation 4.5); (2) for instances to the specialized Dart game Dart_G^k over G (Notation B.3). In any case, for a subset $C \subseteq B$, $x\restriction_C$ denotes the tuple $\langle x\restriction_c \rangle_{c \in C} \in A^C$; for a subset $Y \subseteq X$, $Y\restriction_b := \{y\restriction_b\}_{y \in Y}$ for $b \in B$ and $Y\restriction_C := \{y\restriction_C\}_{y \in Y}$ for $C \subseteq B$.

This work focuses on boolean circuits of fan-in two having a single output gate, and the main concern is their depth complexity, measured by the number of edges on the longest path from an input gate to the output gate (which may be zero), where negation costs no increase in depth.

We fix our notation for directed acyclic graphs below. For brevity, immediate predecessors are called in-neighbors here, and immediate successors are called out-neighbors.

Notation 2.2 (Directed Acyclic Graph)**.** Consider a directed acyclic graph (DAG) $G = (V, E)$. For every vertex $a \in V$, denote its in-neighbors as $\delta^{\text{in}}(a) := \{b \in V : (b, a) \in E\}$ and out-neighbors as $\delta^{\text{out}}(a) := \{b \in V : (a, b) \in E\}$, and in-degree as $\deg^{\text{in}}(a) := |\delta^{\text{in}}(a)|$. For the DAG G, its source vertices are $U := U(G) := \{a \in V : \delta^{\text{in}}(a) = \emptyset\}$ and sink vertices are $W := W(G) := \{a \in V : \delta^{\text{out}}(a) = \emptyset\}$.

3 Equivalence of Pebble Games

We first informally review the three pebble games (§§ 3.1 to 3.3), and then show their equivalence (§§ 3.4 and 3.5).

To avoid confusion with the two-*party* communication games of Karchmer–Wigderson (see § 4.1) or of Raz–McKenzie (called Dart game, see Appendix B.1), this paper refers to *Pebbler* and *Challenger* (or *Colorer*) as the two *players* in a Dymond–Tompa game (or Raz–McKenzie pebble game).

3.1 Dymond–Tompa Game

The following version of the Dymond–Tompa game is needed, where *Pebbler* only pebble one vertex in each round, similar to the variant used in [BCGR92]. Concerning the number of pebbles, this one-pebble-per-round version is clearly equivalent to the original multiple-pebble-per-round version by Dymond and Tompa (by a simulation argument). The informal Definition 3.1 is is formalized as Definition 3.4 in § 3.4. Its pebble cost is the time needed.

Definition 3.1 (Dymond–Tompa Game [DT85])**.** Fix a DAG G. The Dymond–Tompa game (DT_G) over G is a two-player (competitive) game as follows. The two players, *Pebbler* and *Challenger*, alternate to move. The *Pebbler* begins by pebbling a sink vertex of G, which is then challenged by *Challenger*. In all subsequent rounds, *Pebbler* places a pebble on a vertex of G, then *Challenger* either (1) rechallenges the currently challenged vertex; or (2) challenges the vertex pebbled by *Pebbler*. The game is over when *Challenger* challenges $a \in V$, but all in-neighbors of a are pebbled. A game takes h time if *Pebbler* needs h pebble moves to win, against an optimal *Challenger* play.

3.2 Raz–McKenzie Pebble Game

Raz–McKenzie [RM99] employed the following pebble game in their adversarial strategy for proving lower bounds on the depth of monotone circuits. Elias–McKenzie [EM10] initiated the study of the pebble game over different directed acyclic graphs. The informal Definition 3.2 is formalized as Definition 3.15 in § 3.4. Its pebble cost is the time needed.

Definition 3.2 (Raz–McKenzie Pebble Game)**.** Fix a DAG G. The Raz–McKenzie pebble game (RM_G) over G is a two-player (competitive) game as follows. The two players, *Pebbler* and *Colorer*, alternate to move. The *Pebbler* begins by pebbling a sink vertex of G, which is then colored red by *Colorer*. In all subsequent rounds, *Pebbler* places a pebble on a vertex of G, then *Colorer* colors this vertex either (1) as red; or (2) as blue. The game is over when some vertex $a \in V$ is colored red, but all in-neighbors of a are colored blue. A game takes h time if *Pebbler* needs h pebble moves to win, against an optimal *Colorer* play.

3.3 Reversible Pebble Game

Bennett [Ben89] mentioned reversible pebble game as an abstraction for a reversible simulation of irreversible computation. The informal Definition 3.3 is formalized as Definition 3.20. Its pebble cost is the number of pebbles needed.

Definition 3.3 (Reversible Pebble Game)**.** Fix a DAG G. The reversible pebble game over G is a one-player game as follows. Each vertex of G can store at most one pebble, and the game begins with no pebbles on G. In each move, *Pebbler* applies one of the following rules: (1) if all in-neighbors of a are pebbled, *Pebbler* may place a pebble on a (to pebble a); or (2) if all in-neighbors of a are pebbled, *Pebbler* may remove a pebble from a (to unpebble a). The game is over when the sink vertex is pebbled, but all other vertices are unpebbled. A game takes h pebbles if *Pebbler* needs h pebbles to finish the game.

3.4 When Dymond–Tompa meet Raz–McKenzie

This section formalizes the Dymond–Tompa game (§ 3.4.1) and the Raz–McKenzie pebble game (§ 3.4.2), and proves their equivalence (§ 3.4.3).

3.4.1 Dymond–Tompa Game

Definitions 3.4 and 3.5 formalize the intuitive Definition 3.1 for the Dymond–Tompa Game.

Definition 3.4 (Dymond–Tompa Game Tree)**.** Fix a DAG $G = (V, E)$. A *configuration* of the Dymond–Tompa game (DT_G) over G is a tuple $\langle\!\langle P, r, c\rangle\!\rangle$, where $P \subseteq V$ are the pebbled vertices, $r \in P \cup \{\perp\}$ is the vertex just pebbled, and $c \in P$ is the vertex under challenge. The player taking the turn in $\langle\!\langle P, r, c\rangle\!\rangle$ is *Pebbler* if $r = \perp$, and is *Challenger* if $r \in P$.

The *initial configuration* for G is $C_G := \langle\!\langle \{\tau\}, \perp, \tau\rangle\!\rangle$,[22] and the game is *over* in a configuration $\langle\!\langle P, r, c\rangle\!\rangle$ if $r = \perp$ and $\delta^{\text{in}}(c) \subseteq P$. A configuration $C := \langle\!\langle P, r, c\rangle\!\rangle$ *moves* to a configuration $C' := \langle\!\langle P', r', c'\rangle\!\rangle$ (denoted as $C \vdash C'$), if (1) $r = \perp$ and $r' \in V \setminus P$ (*Pebbler* moves in C and then *Challenger* moves in C'),[23] and the game is not over in C and $P' = P \cup \{r'\}$ and $c' = c$; or (2)

[22] Recall that G is assumed to have a unique sink vertex τ.

[23] Note that $r' \in V \setminus P$ in item (1) in the definition of $\vdash$, i. e., *Pebbler* is required to pebble an unpebbled vertex. The game is effectively the same with or without this requirement, since *Challenger* can always rechallenge the last challenged vertex if *Pebbler* repebbles a pebbled vertex, hence an optimal *Pebbler* strategy should obey this requirement. This requirement is added here to avoid working with an infinite game tree, so as to simplify subsequent definitions while not affecting the values of subgames.

$r \in P$ and $r' = \perp$ (*Challenger* moves in C and then *Pebbler* moves in C'), and $c' \in \{c, r\}$ and $P' = P$.

In the Dymond–Tompa game tree ($\textsc{GameTree}_G$) for DT_G, every node is labeled with a configuration. First construct the root node of $\textsc{GameTree}_G$, labeled with the initial configuration C_G. And for any node x labeled with C, for every C' such that $C \vdash C'$, construct a child node x' of x labeled with C'. The game tree is finite since *Pebbler* is required to pebble an unpebbled vertex.[23]

Definition 3.5 (Value of a (Sub)-Game)**.** For a node x on $\textsc{GameTree}_G$, define its value

$$\textsc{Val}(x) := \begin{cases} 1 & \text{if } x \text{ is a leaf node,} \\ \min_{x'\colon \text{child of } x} \textsc{Val}(x') & \text{if } \textit{Pebbler} \text{ moves at internal node } x, \\ 1 + \max_{x'\colon \text{child of } x} \textsc{Val}(x') & \text{if } \textit{Challenger} \text{ moves at internal node } x. \end{cases}$$

Then DT_G takes h time if $\textsc{Val}(\text{root of } \textsc{GameTree}_G) = h$.

Intuitively, an optimal game play should focus only on the effective predecessors $V_c(P)$ of the currently challenged vertex c (Definition 3.6). This is formalized as Lemma 3.8, by an induction on Lemma 3.9.

Definition 3.6 (Effective Predecessors)**.** Relative to any $S \subseteq V$, for vertices a and b in V, define the transitive relation $a \rightsquigarrow_S b$ if there is a directed path (possibly of zero length) from a to b avoiding S, i. e., there exists $\{v_0, v_1, \dots, v_\ell\} \subseteq V \setminus S$ such that $v_0 = a$ and $v_\ell = b$ and $v_i \in \delta^{\text{in}}(v_{i+1})$ for $0 \le i < \ell$. When c is under challenge and P are the pebbled vertices, define the (not necessarily proper) *effective predecessors* of c avoiding P as $V_c(P) := \{a \in V : a \rightsquigarrow_{(P \setminus \{c\})} c\}$.

Proposition 3.7 (Effective Predecessors)**.** *We have the following:*

1. $V_c(P) \cap P = \{c\}$ *when* $c \in P$*;*
2. *If* $a \in V_c(Q)$*, then* $V_a(Q) \subseteq V_c(Q)$*;*
3. *If* $c \in Q \subseteq R$*, then* $V_c(Q) \supseteq V_c(R)$*; and*
4. *If* $c \in Q \subseteq R$ *and* $(R \setminus Q) \cap V_c(Q) = \emptyset$*, then* $V_c(Q) = V_c(R)$*.*

Lemma 3.8 (Predecessors Determine a Subgame)**.** *The value of a subgame depends only on the effective predecessors of the challenged vertex, i. e., if* $V_c(Q) = V_c(R)$*, then* $\textsc{Val}(\langle\!\langle Q, \perp, c\rangle\!\rangle) = \textsc{Val}(\langle\!\langle R, \perp, c\rangle\!\rangle)$*.*[24]

Proof. Let $P = Q \cup \big(V \setminus V_c(Q)\big) = R \cup \big(V \setminus V_c(R)\big)$, then $Q \subseteq P$ and $R \subseteq P$, now do an induction using Lemma 3.9 to show $\textsc{Val}(\langle\!\langle Q, \perp, c\rangle\!\rangle) = \textsc{Val}(\langle\!\langle P, \perp, c\rangle\!\rangle) = \textsc{Val}(\langle\!\langle R, \perp, c\rangle\!\rangle)$. More precisely, recall that for a subset $S \subseteq V$, a sink vertex s of S satisfies $s \in S$ and $\delta^{\text{out}}(s) \cap S = \emptyset$. Enumerate $P \setminus Q =: \{s_1, s_2, \dots, s_\ell\} \subseteq V \setminus \big(Q \cup V_c(Q)\big)$ so that s_i is a sink of S_i where $S_\ell := P$ and $S_i := S_{i+1} \setminus \{s_{i+1}\}$ for $0 \le i < \ell$, and apply Lemma 3.9 to get $\textsc{Val}(\langle\!\langle S_i, \perp, c\rangle\!\rangle) = \textsc{Val}(\langle\!\langle S_{i+1}, \perp, c\rangle\!\rangle)$. ☐

Lemma 3.9 (Predecessors Determine Adjacent Subgames)**.** *If* $R = Q \cup \{q\}$ *for some sink* q *of* $V \setminus \big(Q \cup V_c(Q)\big)$*, then* $\textsc{Val}(\langle\!\langle Q, \perp, c\rangle\!\rangle) = \textsc{Val}(\langle\!\langle R, \perp, c\rangle\!\rangle)$*.*[24]

[24] Clearly the subtree rooted at (and hence the value of) a node x on $\textsc{GameTree}_G$ depends only on the configuration labeled at x, thus it makes sense to talk about the value of a configuration, although in general there can be multiple nodes on $\textsc{GameTree}_G$ labeled with the same configuration.

Proof. Say two *Pebbler* configurations $C_1 := \langle\!\langle P_1, \bot, c_1 \rangle\!\rangle$ and $C_2 := \langle\!\langle P_2, \bot, c_2 \rangle\!\rangle$ form an *adjacent pair* (denoted $\langle C_1, C_2 \rangle$) if $c_1 = c = c_2$ for some $c \in V$ and $P_2 = P_1 \cup \{q\}$ for some sink q of $V \setminus \big(P_1 \cup V_c(P_1)\big)$. In this case $V_c(P_1) = V_c(P_2)$ by Proposition 3.7. For two configurations C and C', say C is a *descendant* of C' (denoted $C \preceq C'$) if there are configurations $\{C_1, \dots, C_\ell\}$, such that $C_{i+1} \vdash C_i$ for $1 \le i < \ell$ and $C_1 = C$ and $C_\ell = C'$.[25] This partial order on configurations induces a partial order on adjacent pairs by $\langle C_1, C_2 \rangle \preceq \langle C_1', C_2' \rangle$ if $C_1 \preceq C_1'$ and $C_2 \preceq C_2'$.

Do an induction following the $\preceq$ order on adjacent pairs $\langle Q, R \rangle$ to show that $\textsc{Val}(Q) = \textsc{Val}(R)$. When $V_c(Q) = V_c(R)$, note that $\delta^{\text{in}}(c) \subseteq Q$ iff $V_c(Q) = \{c\}$ iff $V_c(R) = \{c\}$ iff $\delta^{\text{in}}(c) \subseteq R$, i. e., the game is over in $\langle\!\langle Q, \bot, c \rangle\!\rangle$ iff it is over in $\langle\!\langle R, \bot, c \rangle\!\rangle$. If the game is over, then $\textsc{Val}(\langle\!\langle Q, \bot, c \rangle\!\rangle) = 1 = \textsc{Val}(\langle\!\langle R, \bot, c \rangle\!\rangle)$, establishing the base case. Otherwise, the game is not over. Expand and compare

$$\textsc{Val}(\langle\!\langle Q, \bot, c \rangle\!\rangle) = \min_{r \notin Q} \textsc{Val}(\langle\!\langle Q \cup \{r\}, r, c \rangle\!\rangle) \quad \text{and} \quad \textsc{Val}(\langle\!\langle R, \bot, c \rangle\!\rangle) = \min_{r \notin R} \textsc{Val}(\langle\!\langle R \cup \{r\}, r, c \rangle\!\rangle) \ .$$

For an $r \in V \setminus Q$, there are two cases.

- $r \notin R$: Note that $\langle \langle\!\langle Q \cup \{r\}, \bot, c \rangle\!\rangle, \langle\!\langle R \cup \{r\}, \bot, c \rangle\!\rangle \rangle \prec \langle \langle\!\langle Q, \bot, c \rangle\!\rangle, \langle\!\langle R, \bot, c \rangle\!\rangle \rangle$, and since q is a sink of $V \setminus \big(Q \cup V_c(Q)\big)$ and $q \notin V_r(Q \cup \{r\})$, we have $\langle \langle\!\langle Q \cup \{r\}, \bot, r \rangle\!\rangle, \langle\!\langle R \cup \{r\}, \bot, r \rangle\!\rangle \rangle \prec \langle \langle\!\langle Q, \bot, c \rangle\!\rangle, \langle\!\langle R, \bot, c \rangle\!\rangle \rangle$, hence induction hypothesis gives

$$\begin{aligned} \textsc{Val}(\langle\!\langle Q \cup \{r\}, r, c \rangle\!\rangle) &= 1 + \max\big\{ \ \textsc{Val}(\langle\!\langle Q \cup \{r\}, \bot, r \rangle\!\rangle) \ , \ \textsc{Val}(\langle\!\langle Q \cup \{r\}, \bot, c \rangle\!\rangle) \ \big\} \\ &= 1 + \max\big\{ \ \textsc{Val}(\langle\!\langle R \cup \{r\}, \bot, r \rangle\!\rangle) \ , \ \textsc{Val}(\langle\!\langle R \cup \{r\}, \bot, c \rangle\!\rangle) \ \big\} \\ &= \textsc{Val}(\langle\!\langle R \cup \{r\}, r, c \rangle\!\rangle) \ ; \end{aligned}$$

- $r = q \in R \setminus Q$: Then

$$\begin{aligned} \textsc{Val}(\langle\!\langle Q \cup \{q\}, q, c \rangle\!\rangle) &= \textsc{Val}(\langle\!\langle R, q, c \rangle\!\rangle) \\ &= 1 + \max\big\{ \ \textsc{Val}(\langle\!\langle R, \bot, q \rangle\!\rangle) \ , \ \textsc{Val}(\langle\!\langle R, \bot, c \rangle\!\rangle) \ \big\} \\ &> \textsc{Val}(\langle\!\langle R, \bot, c \rangle\!\rangle) \ . \end{aligned}$$

Now

$$\begin{aligned} \textsc{Val}(\langle\!\langle Q, \bot, c \rangle\!\rangle) &= \min_{r \notin Q} \textsc{Val}(\langle\!\langle Q \cup \{r\}, r, c \rangle\!\rangle) \\ &= \min\big\{ \ \min_{r \notin R} \textsc{Val}(\langle\!\langle Q \cup \{r\}, r, c \rangle\!\rangle) \ , \ \textsc{Val}(\langle\!\langle Q \cup \{q\}, q, c \rangle\!\rangle) \ \big\} \\ &= \textsc{Val}(\langle\!\langle R, \bot, c \rangle\!\rangle) \ . \end{aligned}$$

□

Since the game should only focus on the effective predecessors $V_c(P)$ of the currently challenged vertex c, an optimal game play should go from the sink to the sources of G (Claims 3.11 and 3.13, see Definition 3.10).

Definition 3.10 (Upstream Strategies)**.** Say a strategy for *Pebbler* is *upstream* if *Pebbler* only pebbles an effective predecessor of the currently challenged vertex, and a strategy for *Challenger* is *upstream* if *Challenger* only challenges an effective predecessor of the previously challenged vertex. More precisely, for configurations $C := \langle\!\langle P, r, c \rangle\!\rangle$ and $C' := \langle\!\langle P', r', c' \rangle\!\rangle$, say *$C$ moves upstream to C'* (denoted as $C \vDash\!\!\leftsquigarrow C'$) iff $C \vdash C'$ and if (1) $r = \bot$ (*Pebbler* moves in C) then $r' \in V_c(P)$; or (2) $r \in P$ (*Challenger* moves in C) then $c' \in V_c(P)$. Then a *Pebbler* (resp. *Challenger*) strategy is upstream if every *Pebbler* (resp. *Challenger*) move from C to C' satisfies $C \vDash\!\!\leftsquigarrow C'$.

[25] Hence $C \preceq C'$ if some node labeled with C is a (not necessarily proper) descendant of some node labeled with C' on $\textsc{GameTree}_G$.

Claim 3.11 (Optimal Upstream *Pebbler*)**.** *Any subgame-optimal Pebbler strategy is upstream, i. e., if configurations $C = \langle\!\langle P, \perp, c\rangle\!\rangle$ and C' satisfy $C \vdash C'$ and $\textsc{Val}(C) = \textsc{Val}(C')$,*[24] *then $C \vdash_{\leftsquigarrow} C'$.*

Proof. Consider the *Pebbler* move from $C =: \langle\!\langle P, \perp, c\rangle\!\rangle$ to $C' =: \langle\!\langle P \cup \{r\}, r, c\rangle\!\rangle$ where $r \notin P$ and $r \notin V_c(P)$ (hence $C \vdash C'$ but $C \not\vdash_{\leftsquigarrow} C'$), either (1) challenging r is no worse for *Challenger*, i. e., $C_r := \langle\!\langle P \cup \{r\}, \perp, r\rangle\!\rangle$ has $\textsc{Val}(C_r) \geq \textsc{Val}(C)$, then a subgame-optimal strategy of *Pebbler* would avoid the move from C to C' (since $\textsc{Val}(C') \geq 1 + \textsc{Val}(C_r) > \textsc{Val}(C)$); or (2) C_r is worse for *Challenger*, i. e., $\textsc{Val}(C_r) < \textsc{Val}(C)$, then *Challenger* may choose to rechallenge c by moving to $C_c := \langle\!\langle P \cup \{r\}, \perp, c\rangle\!\rangle$ so that $\textsc{Val}(C_c) = \textsc{Val}(C)$ (by Proposition 3.7 and Lemma 3.8), hence a subgame-optimal strategy of *Pebbler* would avoid the move from C to C' (since $\textsc{Val}(C') \geq 1 + \textsc{Val}(C_c) > \textsc{Val}(C)$). □

Corollary 3.12 (Optimal Upstream *Pebbler*)**.** *If the game is not over in a Pebbler configuration $\langle\!\langle P, \perp, c\rangle\!\rangle$, then $\textsc{Val}(\langle\!\langle P, \perp, c\rangle\!\rangle) = \min_{r \in V_c(P) \setminus P} \textsc{Val}(\langle\!\langle P \cup \{r\}, r, c\rangle\!\rangle)$.*

Claim 3.13 (Optimal Upstream *Challenger*)**.** *There exists an optimal Challenger strategy that is upstream, i. e., if configurations $C = \langle\!\langle P, \perp, c\rangle\!\rangle$ and $C' = \langle\!\langle P \cup \{r\}, r, c\rangle\!\rangle$ satisfy $C \vdash C'$, then there is a Challenger move from C' to C'' with $C' \vdash_{\leftsquigarrow} C''$ and $\textsc{Val}(C'') \geq \textsc{Val}(C) - 1$.*

Proof. If $r \in V_c(P)$, then $C' \vdash C''$ implies $C' \vdash_{\leftsquigarrow} C''$. Now Definition 3.5 gives a C'' with $C' \vdash_{\leftsquigarrow} C''$ and $\textsc{Val}(C'') \geq \textsc{Val}(C') - 1 \geq \textsc{Val}(C) - 1$. Otherwise $r \notin V_c(P)$, then consider $C'' := \langle\!\langle P \cup \{r\}, \perp, c\rangle\!\rangle$. Proposition 3.7 and Lemma 3.8 give $\textsc{Val}(C'') = \textsc{Val}(C)$, and clearly $C' \vdash_{\leftsquigarrow} C''$. □

Proposition 3.14 (Upstream is Monotone)**.** *Consider $C_1 \vdash C_2 \vdash C_3$ where $C_1 =: \langle\!\langle P_1, \perp, c_1\rangle\!\rangle$ and $C_3 =: \langle\!\langle P_3, \perp, c_3\rangle\!\rangle$. If $C_1 \vdash_{\leftsquigarrow} C_2$ or $C_2 \vdash_{\leftsquigarrow} C_3$, then $c_3 \in V_{c_1}(P_1)$ and $V_{c_1}(P_1) \supseteq V_{c_3}(P_3)$.*

3.4.2 Raz–McKenzie Pebble Game

Definitions 3.15 and 3.16 formalize the intuitive Definition 3.2 for the Raz–McKenzie pebble game.

Definition 3.15 (Raz–McKenzie Game Tree)**.** Fix a DAG $G = (V, E)$. A *configuration* of the Raz–McKenzie game (RM_G) over G is a tuple $\langle\!\langle P, r, B\rangle\!\rangle$, where $P \subseteq V$ are the pebbled vertices, $r \in P \cup \{\perp\}$ is the vertex just pebbled, and $B \subset P$ are the blue vertices (and $P \setminus (B \cup \{r\})$ are the red vertices). The player taking the turn in $\langle\!\langle P, r, B\rangle\!\rangle$ is *Pebbler* if $r = \perp$, and is *Colorer* if $r \in P$.

The *initial configuration* for G is $C_G^{\mathsf{RM}} := \langle\!\langle \{\tau\}, \perp, \emptyset\rangle\!\rangle$,[22] and the game is *over* in a configuration $\langle\!\langle P, r, B\rangle\!\rangle$ if $r = \perp$ and some $d \in P \setminus B$ has $\delta^{\mathrm{in}}(d) \subseteq B$. A configuration $C := \langle\!\langle P, r, B\rangle\!\rangle$ *moves* to a configuration $C' := \langle\!\langle P', r', B'\rangle\!\rangle$ (denoted as $C \vdash C'$), if (1) $r = \perp$ and $r' \in V \setminus P$ (*Pebbler* moves in C and then *Colorer* moves in C'),[26] and the game is not over in C and $P' = P \cup \{r'\}$ and $B' = B$; or (2) $r \in P$ and $r' = \perp$ (*Colorer* moves in C and then *Pebbler* moves in C'), and $B \subseteq B' \subseteq B \cup \{r\}$ and $P' = P$.

In the Raz–McKenzie game tree ($\textsc{GameTree}_G^{\mathsf{RM}}$) for RM_G, every node is labeled with a configuration. First construct the root node of $\textsc{GameTree}_G^{\mathsf{RM}}$, labeled with the initial configuration C_G^{RM}. And for any node x labeled with C, for every C' such that $C \vdash C'$, construct a child node x' of x labeled with C'. The game tree is finite since *Pebbler* is required to pebble an unpebbled vertex.[26]

[26]Note that $r' \in V \setminus P$ in item (1) in the definition of $\vdash$, i. e., *Pebbler* is required to pebble an unpebbled vertex. The game is effectively the same with or without this requirement, since *Colorer* can always recolor a vertex with its existing color if *Pebbler* repebbles a pebbled vertex, hence an optimal *Pebbler* strategy should obey this requirement. This requirement is added here to avoid working with an infinite game tree, so as to simplify subsequent definitions while not affecting the values of subgames.

Definition 3.16 (Value of a (Sub)-Game)**.** For a node x on $\textsc{GameTree}_G^{\mathsf{RM}}$, define its value

$$\textsc{Val}(x) := \begin{cases} 1 & \text{if } x \text{ is a leaf node,} \\ \min_{x' \colon \text{child of } x} \textsc{Val}(x') & \text{if } \textit{Pebbler} \text{ moves at internal node } x, \\ 1 + \max_{x' \colon \text{child of } x} \textsc{Val}(x') & \text{if } \textit{Colorer} \text{ moves at internal node } x. \end{cases}$$

Then RM_G takes h time if $\textsc{Val}(\text{root of } \textsc{GameTree}_G^{\mathsf{RM}}) = h$.

3.4.3 Dymond–Tompa equals Raz–McKenzie

Theorem 6 (Dymond–Tompa equals Raz–McKenzie)**.** *For any DAG G, DT_G takes h time iff RM_G takes h time.*

Proof. For the $\Leftarrow$ direction, given an optimal *Colorer* strategy for RM_G, we should construct a *Challenger* strategy for DT_G to make at least h moves. In each move, when c is under challenge, after *Pebbler* pebbles $r \in V \setminus P$ to a configuration $\langle\!\langle P \cup \{r\}, r, c\rangle\!\rangle$ in DT_G, *Challenger* (1) challenges r if $r \in V_c(P)$ and *Colorer* colors r red in response to *Pebbler*; and (2) rechallenges c otherwise. *Challenger* strategy maintains the invariant that c is the only red vertex among its effective predecessors, i. e., $V_c(P) \cap (P \setminus B) = \{c\}$ (by induction on *Challenger* moves). If the game DT_G is over in a configuration $\langle\!\langle P, \perp, c\rangle\!\rangle$, then $\delta^{\text{in}}(c) \subseteq P$. It follows that c is red but all of $\delta^{\text{in}}(c)$ are blue; for otherwise, some $r \in \delta^{\text{in}}(c)$ is red, but then r is colored red by *Colorer* in a round when some d is challenged, and both c and r are effective predecessors of d in that round (recall Proposition 3.14), contradicting the *Challenger* strategy. So the game RM_G is also over.

For the $\Rightarrow$ direction, given an optimal *Challenger* strategy for DT_G, we should construct a *Colorer* strategy for RM_G to make at least h moves. By Claim 3.13, assume that *Challenger* strategy is upstream. In each move, when c is under challenge, after *Pebbler* pebbles $r \in V \setminus P$ to configurations $\langle\!\langle P', r, c\rangle\!\rangle$ in DT_G and $\langle\!\langle P', r, B\rangle\!\rangle$ in RM_G with $P' = P \cup \{r\}$, if (1) *Challenger* responses to *Pebbler* by challenging $r \neq c$ (hence $r \in V_c(P)$), then *Colorer* colors r red; or (2) *Challenger* responses by rechallenging c, then *Colorer* (i) colors r blue unless there are red vertices $d, d' \in P \setminus B$ blocked by making r blue, i. e., $d \notin V_{d'}(B \cup \{r\})$ but $d \in V_{d'}(B)$; in which case (ii) colors r red. *Colorer* strategy maintains the invariant that c is the only red vertex among its effective predecessors, i. e., $V_c(P) \cap (P \setminus B) = \{c\}$; and there is a blue-avoiding path covering all red vertices, i. e., for any $d, d' \in P \setminus B$, $d \in V_{d'}(B)$ or $d' \in V_d(B)$ (by induction on *Colorer* moves). As a result, c is the first red vertex in this blue-avoiding path, i. e., $c \in V_d(B)$ for any $d \in P \setminus B$. If the game RM_G is over in a configuration $\langle\!\langle P, \perp, B\rangle\!\rangle$, then $\delta^{\text{in}}(d) \subseteq B$ for some $d \in P \setminus B$, thus $d = c$ is the vertex under challenge and $\delta^{\text{in}}(c) \subseteq P$, so the game DT_G is also over. □

3.5 When Raz–McKenzie meet Bennett

The Raz–McKenzie pebble game (§ 3.5.1) is connected with the reversible pebble game of Bennett (§ 3.5.2) by a simulation argument in § 3.5.3.

3.5.1 Reformulating Raz–McKenzie Pebble Game

Focusing on the *Pebbler* side of the Raz–McKenzie pebble game and interpreting it as a one-player game, Definition 3.17 and Proposition 3.18 bring the Raz–McKenzie (two-player) pebble game to a form closer to the (one-player) reversible pebble game.

Definition 3.17 (Reduced Configuration)**.** Fix a DAG $G = (V, E)$. A *reduced configuration* of the Raz–McKenzie pebble game (RM_G) over G is a pair $(\!| B, R |\!)$ of blue B and red R vertices

$(B, R \subseteq V)$ which are disjoint $B \cap R = \{\}$. Any reduced configuration $\llparenthesis B, R \rrparenthesis$ corresponds to the *Pebbler* configuration $\langle\!\langle R \cup B, \perp, B \rangle\!\rangle$.

Proposition 3.18 (Value of Reduced Configuration)**.**

$$\textsc{Val}(\llparenthesis B, R \rrparenthesis) = \begin{cases} 1 & \text{if } \exists r \in R \text{ s.t. } \delta^{in}(r) \subseteq B, \\ 1 + \min\limits_{v \in V \setminus (R \cup B)} \max \left\{ \begin{array}{l} \textsc{Val}(\llparenthesis B, R \cup \{v\} \rrparenthesis), \\ \textsc{Val}(\llparenthesis B \cup \{v\}, R \rrparenthesis) \end{array} \right\} & \text{otherwise.} \end{cases}$$

Proposition 3.19 (Monotonicity)**.** *If* $B_1 \subseteq B_2$ *and* $R_1 \subseteq R_2$*, then* $\textsc{Val}(\llparenthesis B_1, R_1 \rrparenthesis) \geq \textsc{Val}(\llparenthesis B_2, R_2 \rrparenthesis)$.

3.5.2 Reversible Pebble Game

Following its usage in proof complexity [BSIW04, AJPU07, HU07, Urq11], the (reversible) black pebble game is parameterized below with two extra sets of vertices S (extending the sources) and T (extending the sinks) in Definition 3.20. By changing S and T as the induction step goes, and by focusing on the progress in pebbling outside of S and T, this parameterization sets up the right recurrence in its translation to and from the Raz–McKenzie pebble game (Lemmas 3.22 and 3.25).

Definition 3.20 (Reversible Pebble Game)**.** Fix a DAG $G = (V, E)$ and two vertex subsets $S, T \subseteq V$ which are disjoint $S \cap T = \{\}$. A *configuration* P in the reversible pebble game ($\mathsf{RP}_{G,S,T}$) is a subset of pebbled vertices $P \subseteq V$. Two configurations P_1 and P_2 are *adjacent* in $\mathsf{RP}_{G,S,T}$ if P_1 and P_2 differ by at most one vertex $v \in V$, all of whose in-neighbors are pebbled or in S, i. e., $P_1 \Delta P_2 \subseteq \{v\}$ where $\delta^{\mathrm{in}}(v) \subseteq P_1 \cup S$ for some $v \in V$ (in this case, equivalently $\delta^{\mathrm{in}}(v) \subseteq P_2 \cup S$). Note that all of S are virtually pebbled, hence referred to as *assuming* S. Say a configuration P *precisely pebbles* a vertex in $T \subseteq V$ *assuming* $S \subseteq V$ if $P \setminus S = \{t\}$ for some vertex $t \in T$. For two configurations P_s and P_t, a *reversible (pebbling) strategy* $\mathcal{P} = \langle P_1, P_2, \ldots, P_\ell \rangle$ *from* P_s *to* P_t *in* $\mathsf{RP}_{G,S,T}$ is a sequence of adjacent configurations, i. e., P_{j-1} is adjacent to P_j assuming S for $1 < j \leq \ell$, such that $P_1 = P_s$ and $P_\ell = P_t$. A *reversible (pebbling) strategy for* $\mathsf{RP}_{G,S,T}$ is a reversible strategy from $\{\}$ to some P_t in $\mathsf{RP}_{G,S,T}$ where P_t precisely pebbles a vertex in T assuming S.

Definition 3.21 (Value of a Configuration)**.** The value of a configuration P is $\textsc{Val}(P) := |P|$ the number of pebbles in P. The value of a reversible strategy $\mathcal{P} := \langle P_1, P_2, \ldots, P_\ell \rangle$ is $\textsc{Val}(\mathcal{P}) := \max_{1 \leq j \leq \ell} \textsc{Val}(P_j)$. The value of the reversible pebble game $\mathsf{RP}_{G,S,T}$ is $\textsc{Val}(\mathsf{RP}_{G,S,T}) := \min_{\mathcal{P}} \textsc{Val}(\mathcal{P})$, where the minimum is over all reversible strategy $\mathcal{P}$ for $\mathsf{RP}_{G,S,T}$ (from $\{\}$ to precisely pebble some vertex in T assuming S in $\mathsf{RP}_{G,S,T}$).

3.5.3 Raz–McKenzie equals Bennett

Lemma 3.22 (Reversible Strategy from Raz–McKenzie Strategy, Induction)**.** *There is a reversible strategy* $\mathcal{P}$ *for* $\mathsf{RP}_{G,B,R}$ *of value* $\textsc{Val}(\mathcal{P}) \leq \textsc{Val}(\llparenthesis B, R \rrparenthesis) =: h$.

Proof. If $h = 1$, then some vertex $r \in R$ has all its in-neighbors $\delta^{\mathrm{in}}(r) \subseteq B$. Now pebble $r \in R$ assuming B, i. e., $\mathcal{P} := \langle \{\}, \{r\} \rangle$, establishing the base case.

If $h > 1$, fix $v \in V \setminus (R \cup B)$ such that

$$\textsc{Val}(\llparenthesis B, R \rrparenthesis) = 1 + \max\left\{ \textsc{Val}(\llparenthesis B, R \cup \{v\} \rrparenthesis), \textsc{Val}(\llparenthesis B \cup \{v\}, R \rrparenthesis) \right\} .$$

Since $\textsc{Val}(\llparenthesis B, R \cup \{v\} \rrparenthesis) < h$, there is a reversible strategy $\mathcal{P}_1 =: \langle P_1, P_2, \ldots, P_\ell \rangle$ for $\mathsf{RP}_{G,B,R \cup \{v\}}$ (i. e., assuming B to precisely pebble a vertex in $R \cup \{v\}$) of value $\textsc{Val}(\mathcal{P}_1) < h$. If a vertex in R is precisely pebbled assuming B ($P_\ell \setminus B = \{r\}$ for some $r \in R$), then we are done $\mathcal{P} := \mathcal{P}_1$. Otherwise, v is precisely pebbled assuming B ($P_\ell \setminus B = \{v\}$). Since $\textsc{Val}(\llparenthesis B \cup \{v\}, R \rrparenthesis) < h$, there is a reversible strategy $\mathcal{P}_2$ for $\mathsf{RP}_{G,B\cup\{v\},R}$ (i. e., assuming $B \cup \{v\}$ to precisely pebble a vertex $r \in R$) of value $\textsc{Val}(\mathcal{P}_2) < h$. Hence run $\mathcal{P}_1$, then run $\mathcal{P}_2$, and finally run $\mathcal{P}_1$ in reverse to forget v. That is, let $\mathcal{P} :=$ the concatenation of $\mathcal{P}_1$, $\mathcal{P}_2 \cup \{v\}$, and $\mathcal{P}_{\overleftarrow{1}} \cup \{r\}$; where $\mathcal{P}_{\overleftarrow{1}} := \langle P_\ell, P_{\ell-1}, \ldots, P_1\rangle$ reverses $\mathcal{P}_1$, and $\mathcal{P}_1 \cup \{r\} := \langle P_1 \cup \{r\}, P_2 \cup \{r\}, \ldots, P_\ell \cup \{r\}\rangle$ denotes the configuration-wise union. Note that $\mathcal{P}$ is a strategy from $\{\}$ to precisely pebble $r \in R$ assuming B. ☐

Lemma 3.23 (Raz–McKenzie Strategy from Reversible Strategy)**.** *Any reversible strategy $\mathcal{P} =: \langle P_1, P_2, \ldots, P_\ell\rangle$ for $\mathsf{RP}_{G,B,R}$ has value $\textsc{Val}(\mathcal{P}) \geq \textsc{Val}(\llparenthesis B, R \rrparenthesis)$.*

Proof. Let $r \in R$ be precisely pebbled assuming B, i. e., $P_\ell \setminus B =: \{r\}$. Without loss of generality $\delta^{\text{in}}(r) \cap R = \emptyset$, for otherwise replace every configuration P_j containing r with $P_j \setminus \{r\} \cup \{r'\}$ for some predecessor r' of r such that $\delta^{\text{in}}(r') \cap R = \emptyset$. Let m be the first time (i. e., least integer) such that r is pebbled since P_m, i. e., $P_b \ni r$ for $m \leq b \leq \ell$. Since $P_1 = \{\}$, $m > 1$. So P_{m-1} differs from P_m by a reversible pebble move to pebble $r \in R \cap P_m$ assuming B. Thus $\delta^{\text{in}}(r) \subseteq P_m \cup B$. Let $\mathcal{P}_1 := \langle P_m, P_{m+1}, \ldots, P_\ell\rangle$ be the strategy since P_m, and $\mathcal{P}_{\overleftarrow{1}} := \langle P_\ell, P_{\ell-1}, \ldots, P_m\rangle$ be its reverse. Note that $\delta^{\text{in}}(r) \subseteq \big(P_m \cap V_R(B)\big) \cup B$ (see Definition 3.24). Apply Lemma 3.25 on $\mathcal{P}_{\overleftarrow{1}} \cap V_R(B)$, where $\mathcal{P}_{\overleftarrow{1}} \cap V_R(B) := \langle P_\ell \cap V_R(B), P_{\ell-1} \cap V_R(B), \ldots, P_m \cap V_R(B)\rangle$ is its configuration-wise intersection with $V_R(B)$, to get P_b with $m \leq b \leq \ell$ satisfying the second inequality in

$$|P_b| \geq \big|\big(P_b \cap V_R(B)\big) \setminus B\big| + 1 \geq \textsc{Val}(\llparenthesis B, R \rrparenthesis),$$

since $P_\ell \cap V_R(B) = \emptyset$ (thus $\tilde{P} = B$ in Lemma 3.25), and $r \in P_b \setminus V_R(B)$ gives the first inequality. ☐

The following is a pebbling argument by induction, with a twist in using the right 'potential function' and the correct order for induction. First, since we are interested in pebbling R, it suffices to restrict attention to predecessors of R in the pebbling strategy. Moreover, since B is assumed, further restrict attention to those pebbling moves outside of B. The region of interest is denoted $V_R(B)$ below. The induction step is applied to the pebbling move P_m where the first vertex (denoted v below) is remembered till the end (i. e., P_ℓ) in the region $V_R(B)$ of interest. By further restricting attention to $V \setminus \{v\}$ (in the Pebbling Case below) or to $V_v(v)$ (in the Unpebbling Case below), it ensures the technical condition that B and R are disjoint when applying the induction hypothesis (as witnessed by the support of a strategy).

Definition 3.24 (Predecessors, Support)**.** Fix a DAG $G = (V, E)$. Say $u \in V$ is a (not necessarily proper) *predecessor* of $v \in V$ if there is a directed path (possibly of zero length) from u to v.[27] Denote the predecessors by $V_v := \{u\colon u \text{ is a predecessor of } v\}$ for $v \in V$, and $V_R := \big(\bigcup_{r\in R} V_r\big)$ for $R \subseteq V$. Define the *predecessors of R relative to B* as $V_R(B) := V_R \setminus (R \cup B)$. As a shorthand, denote $V_v(v) := V_{\{v\}}(\{v\})$ as the *proper* predecessors of v.

For $U \subseteq V$, say a configuration P is *U-supported* if $P \subseteq U$, and say a strategy $\mathcal{P} := \langle P_1, P_2, \ldots, P_\ell\rangle$ is *U-supported* if each P_j is U-supported for $1 \leq j \leq \ell$.

Lemma 3.25 (Raz–McKenzie Strategy from Reversible Strategy, Induction)**.** *Any $V_R(B)$-supported reversible strategy $\mathcal{P} =: \langle P_1, P_2, \ldots, P_\ell\rangle$ in $\mathsf{RP}_{G,B,R}$ where $\delta^{in}(r) \subseteq P_\ell \cup B$ for some $r \in R$, has a configuration P_b for some $1 \leq b \leq \ell$, so that $\big|P_b \setminus \tilde{P}\big| + 1 \geq \textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis)$, where $\tilde{P} := \big(\bigcap_{1\leq j\leq\ell} P_j\big) \cup B$.*

[27]Hence the relation of predecessor is the reflexive transitive closure of the relation of immediate predecessor (in-neighbor).

Proof. Decrease ℓ if necessary, let ℓ be the first time on $\mathcal{P}$ (i. e., least integer) so that $\delta^{\text{in}}(r) \subseteq P_\ell \cup B$ for some $r \in R$. If $\delta^{\text{in}}(r) \subseteq \tilde{P}$, then $\textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis) = 1$, so any configuration on $\mathcal{P}$ would do. Otherwise, $\delta^{\text{in}}(r) \nsubseteq \tilde{P}$. Let $\tilde{P}_i := \left(\bigcap_{i \le j \le \ell} P_j\right) \cup B$ be the set of vertices remembered since configuration P_i assuming B. Now $\delta^{\text{in}}(r) \nsubseteq \tilde{P} = \tilde{P}_1$ and $\delta^{\text{in}}(r) \subseteq \tilde{P}_\ell$, and $\tilde{P}_{j-1} \subseteq \tilde{P}_j$ for $1 < j \le \ell$. Let $m := \operatorname{argmin}\{1 < j \le \ell \colon \tilde{P}_1 \ne \tilde{P}_j\}$ indexes the earliest configuration so that $\tilde{P}_{m-1} \subset \tilde{P}_m$, and let $v \in \tilde{P}_m \setminus \tilde{P}_{m-1} = \tilde{P}_m \setminus \tilde{P}_1$ be the first vertex remembered till the end. Let $\mathcal{P}_1 := \langle P_m, P_{m+1}, \ldots, P_\ell \rangle$ be the strategy since P_m, which is shorter than $\mathcal{P}$.[28] Now for any P_b on $\mathcal{P}_1$ (i. e., $m \le b \le \ell$),

$$|P_b \setminus \tilde{P}_1| \ge |P_b \setminus \tilde{P}_m| + 1 \ , \tag{1}$$

since $v \in \left(P_b \setminus \tilde{P}_1\right) \setminus \left(P_b \setminus \tilde{P}_m\right)$. Note that $\tilde{P}_m = \tilde{P}_1 \cup \{v\} = \tilde{P} \cup \{v\}$.

Clearly $v \notin \tilde{P}$. Also, v is pebbled by a reversible pebble move at P_m, hence $\delta^{\text{in}}(v) \subseteq P_m \cup B$. It follows that $v \notin R$; for otherwise $v \in R$, either it contradicts the minimality of ℓ, or some vertex before v is pebbled till v is pebbled, contradicting the minimality of m. By the recurrence of $\textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis)$ (Proposition 3.18), at least one of the following is true.

- (Pebbling Case)
$$\textsc{Val}(\llparenthesis \tilde{P} \cup \{v\}, R \rrparenthesis) + 1 \ge \textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis) \ . \tag{2}$$
Note that $V_R(B \cup \{v\}) = V_R(B) \setminus \{v\}$. Hence $\mathcal{P}_2 := \mathcal{P}_1 \setminus \{v\} := \langle P_m \setminus \{v\}, P_{m+1} \setminus \{v\}, \ldots, P_\ell \setminus \{v\} \rangle$ is $V_R(B \cup \{v\})$-supported. Now $\tilde{P}_m = \left(\bigcap_{m \le j \le \ell} P_j \setminus \{v\}\right) \cup B \cup \{v\}$. The induction hypothesis on $\mathcal{P}_2$ in $\mathsf{RP}_{G, B \cup \{v\}, R}$ gives a P_b (on $\mathcal{P}_1$) satisfying the inequality in
$$|P_b \setminus \tilde{P}_m| + 1 = \left|\left(P_b \setminus \{v\}\right) \setminus \tilde{P}_m\right| + 1 \ge \textsc{Val}(\llparenthesis \tilde{P}_m, R \rrparenthesis) = \textsc{Val}(\llparenthesis \tilde{P} \cup \{v\}, R \rrparenthesis) \ . \tag{3}$$
Finally $|P_b \setminus \tilde{P}| + 1 \ge \textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis)$ by Inequalities 1 to 3.

- (Unpebbling Case)
$$\textsc{Val}(\llparenthesis \tilde{P}, R \cup \{v\} \rrparenthesis) + 1 \ge \textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis) \ . \tag{4}$$
In fact, $\delta^{\text{in}}(v) \subseteq (P_m \cap V_v(v)) \cup B$. Let $\mathcal{P}_{\overleftarrow{1}} := \langle P_\ell, P_{\ell-1}, \ldots, P_m \rangle$ be the reverse of $\mathcal{P}_1$, and $\mathcal{P}_2 := \mathcal{P}_{\overleftarrow{1}} \cap V_v(v) := \langle P_\ell \cap V_v(v), P_{\ell-1} \cap V_v(v), \ldots, P_m \cap V_v(v) \rangle$ be its configuration-wise intersection. Then $\mathcal{P}_2$ is $V_{R \cup \{v\}}(B)$-supported, and also $V_v(v)$-supported. Let $\tilde{P}' := \left(\bigcap_{m \le j \le \ell} P_j \cap V_v(v)\right) \cup B \subseteq \tilde{P}_m \setminus \{v\} = \tilde{P}$. The induction hypothesis on $\mathcal{P}_2$ in $\mathsf{RP}_{G, B, R \cup \{v\}}$ gives a P_b (on $\mathcal{P}_1$) satisfying the first inequality in
$$|P_b \cap V_v(v) \setminus \tilde{P}'| + 1 \ge \textsc{Val}(\llparenthesis \tilde{P}', R \cup \{v\} \rrparenthesis) \ge \textsc{Val}(\llparenthesis \tilde{P}, R \cup \{v\} \rrparenthesis) \ , \tag{5}$$
where the last inequality follows from monotonicity (Proposition 3.19). Note that $P_b \cap V_v(v) \setminus \tilde{P}' \subseteq P_b \setminus \tilde{P}_m$, since $\tilde{P}_m \cap V_v(v) \subseteq \tilde{P}'$. Finally $|P_b \setminus \tilde{P}| + 1 \ge \textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis)$ by Inequalities 1, 4 and 5. ☐

Corollary 3.26 (Raz–McKenzie equals Bennett)**.** *For any DAG G, subsets $R, B \subseteq V$ which are disjoint $R \cap B = \{\}$, we have* $\textsc{Val}(\llparenthesis B, R \rrparenthesis) = \textsc{Val}(\mathsf{RP}_{G,B,R})$.

Proof. By Lemmas 3.22 and 3.23. ☐

Theorem 7 (Raz–McKenzie equals Bennett)**.** *For any DAG G with a unique sink τ, we have* $\textsc{Val}(\llparenthesis \{\}, \{\tau\} \rrparenthesis) = \textsc{Val}(\mathsf{RP}_{G,\{\},\{\tau\}})$.

[28] Formally, the double induction argument does an outer induction on the length of $\mathcal{P}$, then an inner induction on $\textsc{Val}(\llparenthesis \tilde{P}, R \rrparenthesis)$.

4 DAG Evaluation Problem

This section studies the DAG evaluation problem. We define below the computational problem BDEP^k_G, the boolean version of the DAG evaluation problem of bit-length k over G. § 4.1 recalls the two-party communication game of Karchmer–Wigderson, § 4.2 introduces two classes of circuits with restricted computational semantics for BDEP^k_G, § 4.3 proves an upper bound as Theorem 9, § 4.4 connects the two-player pebble game of Raz–McKenzie with the two-party communication game of Karchmer–Wigderson, and § 4.5 proves a lower bound as Theorem 10.

The following computational problem naturally generalizes the Tree Evaluation Problem [CMW⁺12] to any directed acyclic graph G. This problem can be seen as a parameterized version of the P-complete circuit evaluation problem. By studying a slice of the problem (for a fixed graph G and constant k), we can focus on the combinatorics of the 'flow of values' over the graph.

Definition 4.1 (DAG Evaluation Problem over G)**.** Consider a DAG G and a bit-length parameter $k \in \mathbb{N}$. Denote the set of k-bit strings as $\{0,1\}^k \cong [K]$, where $K := 2^k$. The DAG Evaluation Problem over G (DEP^k_G) is specified by the following.

Input For every vertex $a \in V$, there is a function $t_a\colon [K]^{\delta^{\text{in}}(a)} \to [K]$.[29] The input to DEP^k_G enumerates the n bits of $\langle t_a \rangle_{a\in V}$ as n boolean variables where $n := k \sum_{a\in V} K^{\deg^{\text{in}}(a)}$.

'Computation' Define inductively the values $\langle v_a \rangle_{a\in V} \in [K]^V$ by $v_a := t_a\big(v\restriction_{\delta^{\text{in}}(a)}\big) \in [K]$ for $a \in V$. That is, the value v_a is the function t_a applied to the values at the in-neighbors of a.[29]

Output The output of DEP^k_G is the tuple of values $\langle v_w \rangle_{w\in W} \in [K]^W$.

Using terminologies of database systems, at every vertex $a \in V$, there is a table t_a whose dimension is the number of in-neighbors of a. The values at in-neighbors of a indexes the relevant entry in t_a, and we are interested in computing the values at the sinks.

Henceforth, without loss of generality, focus on DAGs with exactly one sink vertex τ. The interest is in the boolean circuit depth complexity of computing a decision version of DEP^k_G (as opposed to a $[K]$-valued function).

Definition 4.2 (Boolean DAG Evaluation Problem)**.** Fix a non-constant boolean function σ on k-bit strings $\sigma\colon \{0,1\}^k \to \{0,1\}$, say the zeroth bit $\sigma(s) := s\restriction_0$ for $s \in \{0,1\}^k$.[30] The Boolean DAG Evaluation Problem (BDEP^k_G) seeks to compute $\sigma(v_\tau)$.

4.1 Karchmer–Wigderson Game

Boolean circuit depth complexity is studied here via the (co-operative) communication game of Karchmer and Wigderson [KW90]. Recall that given a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ with promises $Y \subseteq f^{-1}(1)$ and $N \subseteq f^{-1}(0)$, the Karchmer–Wigderson game ($\mathsf{KW}_{Y,N}$) is a communication game between two parties defined as follows: Party 1 (the Yes party) is given a promised Yes instance $x \in Y$, Party 0 (the No party) is given a promised No instance $y \in N$, and they communicate to locate a bit position $i \in [n]$ where the inputs differ (i. e., $x_i \neq y_i$). (So the communication protocols are computing relations rather than functions.) And the Karchmer–Wigderson game for a boolean function f is $\mathsf{KW}_f := \mathsf{KW}_{f^{-1}(1),f^{-1}(0)}$. Karchmer and Wigderson observed that the communication complexity captures exactly the circuit depth.[31]

[29] Note that for a source vertex $a \in U$, its function t_a degenerates to have a domain of $[K]^{\emptyset}$, hence the function $t_a \in [K]$ can be treated as a k-bit string. Thus its value v_a is just its function $t_a \in 2^{[K]}$ treated as a k bit-string.

[30] All non-constant boolean functions are equivalent with respect to the (restricted) lower bounds in this work. The zeroth bit is chosen here since its computation is trivial, i. e., takes no extra depth.

[31] Also observed independently by Yannakakis and was implicit in [KPPY84], see [KW90].

Theorem 8 (Karchmer–Wigderson [KW90, Theorem 2.1][31])**.** *The depth complexity of f on boolean circuits is exactly the communication complexity of KW_f.*

Notation 4.3 (Admissible Inputs)**.** Consider the protocol Π (as a rooted binary tree) and a node $g \in \Pi$. Denote $f_g^{-1}(1)$ as the set of inputs that can be given to Party 1 (the YES party) at g and $f_g^{-1}(0)$ as the set of inputs that can be given to Party 0 (the NO party). That is, the combinatorial rectangle associated with the node g is $f_g^{-1}(1) \times f_g^{-1}(0)$.

Notation 4.4 (Output Node)**.** Given an instance $(x, y) \in f^{-1}(1) \times f^{-1}(0)$ and a protocol Π, denote $\Pi(x, y)$ as the output node (rather than just the value) after running the protocol Π on the instance.

4.2 Thrifty and Output-Relevant Circuits

This subsection introduces two families of circuits with restricted computational semantics for BDEP_G^k: thrifty circuits and output-relevant circuits.

When concerning depth complexity, a circuit can be assumed to be a formula without loss of generality. Then a boolean formula $\mathcal{C}$ is isomorphic to a corresponding communication protocol Π (denoted $\mathcal{C} \equiv \Pi$) not only graph-theoretically (as a rooted binary tree), but also *computationally* (subsets of YES and NO instances match the combinatorial rectangles, i. e., for every $g \in \mathcal{C} \equiv \Pi$ under the graph isomorphism, any input $x \in f_g^{-1}(1)$ evaluates to 1 at gate $g \in \mathcal{C}$ and any input $y \in f_g^{-1}(0)$ evaluates to 0 at gate g). Therefore certain computational notions for a formula (or a circuit) $\mathcal{C}$ can equivalently be defined over the communication protocol Π under the Karchmer–Wigderson correspondence $\equiv$, as is done below for the notions of thrifty circuits and output-relevant circuits.

Intuitively, a circuit for BDEP_G^k is thrifty if its computation depends only on the values v_a, but not on other irrelevant bits (variables) of the functions t_a (Definition 4.7), as an analogue of thrifty branching programs [CMW+12]; and a circuit for BDEP_G^k is output-relevant if it only outputs relevant bits (variables), similar to the players who only output leaves of the universal composition relation [KRW95, §6]. Note that after taking away the output-relevant restriction, a communication game for BDEP_G^k is a proper Karchmer–Wigderson game (so that it corresponds properly to circuit depth). This is *not* the case for the universal composition relation.

Notation 4.5 (Values)**.** For an input $x \in \{0,1\}^n$ to BDEP_G^k, denote $x{\restriction}_{v_a}$ as the v_a value of x (see Definition 4.1). As a shorthand, write $x{\restriction}_a$ for $x{\restriction}_{v_a}$, and $x{\restriction}_S$ for $\langle x{\restriction}_a \rangle_{a \in S}$ when $S \subseteq V$.

Definition 4.6 (Thrifty Protocols and Circuits)**.** A protocol Π for $\mathsf{KW}_{Y,N}$ is *thrifty* where $Y \subseteq f^{-1}(1)$ and $N \subseteq f^{-1}(0)$ for $f := \mathsf{BDEP}_G^k$, if for any pair of promised YES instances $x, x' \in Y$, and any pair of promised NO instances $y, y' \in N$, such that $x{\restriction}_V = x'{\restriction}_V$ and $y{\restriction}_V = y'{\restriction}_V$, we have $\Pi(x, y) = \Pi(x', y')$. A circuit $\mathcal{C}$ for f is *thrifty*, if there is a thrifty protocol Π for KW_f isomorphic to (the formula equivalent to) $\mathcal{C}$ (i. e., $\mathcal{C} \equiv \Pi$).

Definition 4.7 (Relevant Bits)**.** For an input $x \in \{0,1\}^n$ to BDEP_G^k, an input bit (variable) is *relevant* to x if (1) it is a variable specifying the $\langle v_{a'} \rangle_{a' \in \delta^{\text{in}}(a)}$ entry of t_a for a vertex $a \in V$; or equivalently (2) $x'{\restriction}_V \neq x{\restriction}_V$ where x' and x differ only on that bit.

Definition 4.8 (Output-Relevant Protocols and Circuits)**.** A protocol Π for $\mathsf{KW}_{Y,N}$ is *output-relevant* where $Y \subseteq f^{-1}(1)$ and $N \subseteq f^{-1}(0)$ for $f := \mathsf{BDEP}_G^k$, if for any $(x, y) \in Y \times N$, the node $\Pi(x, y)$ outputs a bit (position) relevant to x and relevant to y. A circuit $\mathcal{C}$ for f is *output-relevant*, if there is an output-relevant protocol Π for KW_f isomorphic to (the formula equivalent to) $\mathcal{C}$ (i. e., $\mathcal{C} \equiv \Pi$).

Remark 4.9 (Relevant Outputs as Certificates). Recall that the depth of a decision tree depends on the certificate complexity, where a *certificate* for a particular input $x \in \{0,1\}^n$ is a subset of bits of x sufficient to witness the membership/non-membership of x in a language. For both the Dymond–Tompa game (in particular the interpreted variant [VT89]) and the Karchmer–Wigderson game [KW90], it is of interest to efficiently pack certificates (for different YES/NO-instances) into (the leaves of) a shallow 'winning strategy' or (the output nodes of) a shallow protocol. And (the alternation in) the minimization of depth in both games can be modeled by two competing provers, who present bits of the certificates to witness membership/non-membership.

Specializing to the computational problem of BDEP_G^k, an efficient certificate for a particular input $x \in \{0,1\}^n$ should contain precisely the bits of the values relevant to x (at least when k is large, because a certificate containing a full row or column in a table is expensive). This combinatorial consideration motivates the definition of output-relevant circuits.

Proposition 4.10 (Thrifty is Relevant)**.** *For $f := \mathsf{BDEP}_G^k$ and $Y \times N \subseteq f^{-1}(1) \times f^{-1}(0)$, a correct protocol Π for $\mathsf{KW}_{Y,N}$ (and hence a correct circuit $\mathcal{C}$ for f), if thrifty, is output-relevant.*

Proof. If Π for $\mathsf{KW}_{Y,N}$ is *not* output-relevant, there is an instance $(x,y) \in Y \times N$ such that $\Pi(x,y)$ outputs a bit position $i \in [n]$ not relevant to (say) x. Flip that bit in x to get x', then $x'_i \neq x_i$ and $x\restriction_V = x'\restriction_V$. If Π is thrifty, $\Pi(x,y) = \Pi(x',y)$, but then the protocol is incorrect on the instance (x,y) or on (x',y), since either $x_i = y_i$ or $x'_i = y_i$. ☐

4.3 Upper Bound for Evaluation

Theorem 9 implements a strategy for the Dymond–Tompa game DT_G as a circuit for the evaluation problem BDEP_G^k.

Theorem 9 (Upper Bound for Evaluation)**.** *For any directed acyclic graph G whose Dymond–Tompa game takes h time, there is a (uniform) thrifty circuit $\mathcal{C}$ computing BDEP_G^k of depth $(h-1)\big(k + \lceil \log_2(k+1) \rceil\big) = O(hk)$.*

Proof. Apply Lemma 4.12 on $\langle\!\langle \{\tau\}, \perp, \tau \rangle\!\rangle$, $\alpha \in [K]^{\emptyset}$, $j = 0$, $b = 1$. ☐

Definition 4.11 (Bit Equality)**.** Let $\beta\colon \{0,1\}^k \times [k] \times \{0,1\} \to \{0,1\}$ be the *bit-equality* function $\beta(z,j,b) := z\restriction_j \oplus b \oplus 1$, where $\oplus$ denotes addition mod 2.

We recall Definitions 3.4 and 3.5 and Footnote 24, from § 3.4.

Lemma 4.12 (Upper Bound for Evaluation, Induction)**.** *For any configuration $\langle\!\langle P, \perp, c \rangle\!\rangle$ of DT_G with $\textsc{Val}(\langle\!\langle P, \perp, c \rangle\!\rangle) =: h$, any values $\alpha \in [K]^{P \setminus \{c\}}$ on $P \setminus \{c\}$, any $j \in [k]$, $b \in \{0,1\}$, there is a (uniform) thrifty circuit $\mathcal{C}$ for BDEP_G^k of depth $(h-1)\big(k + \lceil \log_2(k+1) \rceil\big)$, so that any $x \in \{0,1\}^n$ with $x\restriction_{P \setminus \{c\}} = \alpha$ satisfies $\mathcal{C}(x) = \beta(x\restriction_{v_c}, j, b)$.*

Proof. Since negation does not increase depth, assume $b = 1$. If $h = 1$, then $\delta^{\text{in}}(c) \subseteq P$, hence α contains all v_a for $a \in \delta^{\text{in}}(c)$. Now the input gate at the j^{th} position of the $\alpha\restriction_{\delta^{\text{in}}(c)}$ entry of t_a is a circuit $\mathcal{C}$ of depth zero satisfying the conditions, establishing the base case. If $h > 1$, let $r \in V_c(P) \setminus P$ be such that $\max\{\textsc{Val}(C_L), \textsc{Val}(C_R)\} < h$ where $C_L := \langle\!\langle P', \perp, r \rangle\!\rangle$ and $C_R := \langle\!\langle P', \perp, c \rangle\!\rangle$ for $P' := P \cup \{r\}$ (Corollary 3.12). Let $C_\Lambda := \langle\!\langle P' \setminus \{c\}, \perp, r \rangle\!\rangle$, then $\textsc{Val}(C_\Lambda) = \textsc{Val}(C_L)$ by Lemma 3.8. Consider a circuit $\mathcal{C}$ constructed as follows: for every $v \in [K] \cong \{0,1\}^k$, let $\alpha_v \in [K]^{P' \setminus \{c\}}$ be such that $\alpha_v\restriction_{P \setminus \{c\}} = \alpha$ and $\alpha_v\restriction_r = v$. For any $i \in [k]$, induction hypothesis on $\langle C_\Lambda, \alpha, i, v\restriction_i \rangle$ gives a circuit $\mathcal{C}_\Lambda^{v,i}$, and induction hypothesis on $\langle C_R, \alpha_v, j, b \rangle$ gives a circuit $\mathcal{C}_R^v$, satisfying the conditions. Construct $\mathcal{C} := \bigvee_{v \in [K]} \big(\mathcal{C}_R^v \wedge \bigwedge_{i \in [k]} \mathcal{C}_\Lambda^{v,i}\big)$, then $\mathrm{depth}(\mathcal{C}) \le \max_{v\in[K]}\{\mathrm{depth}(\mathcal{C}^v_R), \max_{i\in[k]}\{\mathrm{depth}(\mathcal{C}^{v,i}_\Lambda)\}\} + (k + \lceil\log_2(k+1)\rceil)$, and for x with $x\restriction_{P\setminus\{c\}} = \alpha$, $\mathcal{C}(x) = \mathcal{C}^v_R(x)$ for $v := x\restriction_r$. $\square$

4.4 Adversary Argument: when Raz–McKenzie meet Karchmer–Wigderson

Our lower bounds are based on the extension by Raz–McKenzie [RM99] of the adversary argument by Edmonds–Impagliazzo–Rudich–Sgall [EIRS01]. We construct below an interface between the Karchmer–Wigderson (communication) game and the Raz–McKenzie (pebble) game. Note that there is no direct mapping between the two parties in the Karchmer–Wigderson (co-operative) game and the two players in the Raz–McKenzie (competitive) pebble game: the interface between the two games is not straightforward. Also, unlike the case for monotone circuits where it is possible to abstract away the adversary argument as a communication game (called Dart), it appears necessary in the non-monotone case to directly run the adversary argument over the circuit.

Fix a DAG G whose Raz–McKenzie pebble game takes h time, and consider an output-relevant protocol solving BDEP^k_G. It will be shown that the communication game of Karchmer–Wigderson for the evaluation problem $\mathsf{KW}_{\mathsf{BDEP}^k_G}$ must take $\Omega(hk)$ bits of communication for an output-relevant protocol. Recall that the two parties in a communication game want to locate a bit where their inputs differ. Intuitively, for an adversary to foil the two parties, the adversary wants to achieve two conflicting goals: (1) to provide a pair of inputs satisfying the promise of being different; and (2) to hide the difference of a particular input pair among many input pairs, so that the difference is hard for the parties to locate. For hiding the difference, the adversary would maintain a symmetry between the two parties, so that the many input pairs they get look similar (called *same* below). To delay the discovery of the difference (called *different* below) by the two parties, the adversary escalates the decision tree complexity of the Raz–McKenzie pebble game to the communication complexity. For output-relevant protocols, it suffices for the adversary to hide the difference locally, so that no *different* vertices have all its in-neighbors *same* (see Lemma 4.16).

We come up with an adversary (for the Karchmer–Wigderson game) that does the following (to play the Raz–McKenzie pebble game for h moves): she keeps track of a set A of alive vertices over G, and also a set C of common values over (effectively) A (i. e., v_a for $a \in A$) that can be given to both parties (Notation 4.3). The adversary maintains the symmetry between the parties for all the values over A (by keeping C large) until she is forced to kill some vertex in A. Whenever she kills a vertex, she makes a move for *Pebbler* to pebble the newly killed vertex, and then makes a move for *Colorer* under the optimal strategy against *Pebbler*. The parties must spend $\Omega(k)$ bits of communication on average to force the adversary to kill a vertex. And the adversary against an output-relevant protocol is still in good shape unless h vertices are dead (pebbled), because G takes h time to play the Raz–McKenzie pebble game.

A bit more precisely, consider the product space $X := [K]^V$ with $V := V(G)$, interpreted as the set of possible values given to the two parties. At every node g of the protocol Π, any vertex $a \in V$ is either *same* (under symmetry) or *different* (symmetry is broken) for the two parties. Denote $S := S_g \subseteq V$ as the set of *same* vertices, and $D := D_g := V \setminus S$ as the set of *different* vertices at the node g.[32] The meaning of *same* and *different* vertices is as follows (Definition 4.13): for every *different* vertex $d \in D$, there are two non-empty sets $P^0_d, P^1_d \subset X\restriction_d$ of disjoint promised values ($P^0_d \cap P^1_d = \emptyset$), such that P^1_d are some values at d that can be given to Party 1, and P^0_d are some values at d that can be given to Party 0; and there is a set $C := C_g \subseteq X\restriction_S$ of common values over S that can be given to both parties. In addition, a sub-rectangle $Y \times N := Y_g \times N_g \subseteq f_g^{-1}(1) \times f_g^{-1}(0)$ will be associated to a node g.

[32]Formally, the sets Y, N, S, D, C, and A can be different for different gate g, but for cleaner notation we may drop the reference to a gate g when it is clear from the context.

Definition 4.13 (Coherent Data)**.** The data $\langle Y, N, S, D, C, \langle P_d^0, P_d^1\rangle_{d\in D}\rangle$, where $D = V \setminus S$ and $C \subseteq X{\restriction}_S$, is *coherent* at a node g if (i) $Y \times N \subseteq f_g^{-1}(1) \times f_g^{-1}(0)$; and (ii) for any common value $c \in C$, there are YES instance $x \in Y$ and NO instance $y \in N$, so that they agree with c over S (i. e., $x{\restriction}_S = c = y{\restriction}_S$), and are as promised over D (i. e., for any *different* vertex $d \in D$, we have $x{\restriction}_d \in P_d^1$ and $y{\restriction}_d \in P_d^0$).

It will be shown in § 4.5 how the adversary maintains the data $\langle Y, N, S, D, C, \langle P_d^1, P_d^0\rangle_{d\in D}\rangle$ at different nodes g of Π. Consider the subset of *same* vertices whose values have high entropy under C, and call them alive.

Definition 4.14 (Alive Vertices)**.** A vertex $a \in V$ is said to be alive (under C) if $\textsc{AveDeg}_a(C) \geq 8 \cdot K^{19/20}$ (see Definition A.1). Let $A := A_g \subseteq S$ be the set of alive vertices (at node g).

The idea is that, if the *same* vertices S, *different* vertices D, and the alive vertices A form a safe configuration (Definition 4.15) and the data is coherent (Definition 4.13) at a node g, then the adversary is in good shape at g: namely, node g cannot be an output node of the protocol Π, and the two parties need to continue their communication (Lemma 4.16). Note that when $\langle S, D, A\rangle$ is in a safe configuration at a node g, A is non-empty, hence $|C{\restriction}_a| \geq \textsc{AveDeg}_a(C) \gg 0$ for any $a \in A$ and C is non-empty.

Definition 4.15 (Safe Configuration)**.** The triple $\langle S, D, A\rangle$ with $\emptyset \subset A \subseteq S \subseteq V$ and $D = V \setminus S$ is said to be in a safe configuration if every *different* vertex $d \in D$ has at least one in-neighbor $d' \in \delta^{\text{in}}(d)$ such that $d' \in D \cup A$ is *different* or alive.

Lemma 4.16 (Adversary is in Good Shape)**.** *Consider a correct, output-relevant protocol* Π *for* KW_f *where* $f := \mathsf{BDEP}_G^k$*.* ***If*** *at a node* g *of the protocol* Π*, there are sets* $Y \subseteq f_g^{-1}(1)$*,* $N \subseteq f_g^{-1}(0)$*,* $S \subseteq V$ *(same),* $D := V \setminus S$ *(different), and common values* $C \subsetneq X{\restriction}_S$ *over the same vertices* S*, and for any different vertex* $d \in D$*, there are non-empty sets* $P_d^0, P_d^1 \subset X{\restriction}_d$ *of disjoint promised values* ($P_d^0 \cap P_d^1 = \emptyset$) *at* d*, such that (1)* $\langle S, D, A\rangle$ *forms a safe configuration where* A *are the alive vertices (under* C*); and (2) the data* $\langle Y, N, S, D, C, \langle P_d^0, P_d^1\rangle_{d\in D}\rangle$ *is coherent at* g*,* ***then*** *the node* g *cannot be an output node of* Π*.*

Proof. Assume that g is a leaf node of Π outputting a bit position $i \in [n]$. The bit position i specifies a variable of t_a for some $a \in V$. Now consider separately whether a is *same* or *different*.

- $a \in S$ is *same*: since C is non-empty, pick any $c \in C$, and by coherence there is an instance $(x, y) \in f_g^{-1}(1) \times f_g^{-1}(0)$ so that $x{\restriction}_S = c = y{\restriction}_S$, hence $x{\restriction}_a = y{\restriction}_a$. If Π is output-relevant, i is a bit (position) relevant to both x and y, so the relevant entries of x and y are the same at a (i. e., $x{\restriction}_{\delta^{\text{in}}(a)} = y{\restriction}_{\delta^{\text{in}}(a)}$, see item (1) of Definition 4.7) and $x_i = y_i$, so Π cannot be correct.

- $a \in D$ is *different*: since $\langle S, D, A\rangle$ forms a safe configuration, a has an in-neighbor $a' \in \delta^{\text{in}}(a)$ which is *different* or alive.
  - $a' \in D$ is *different*: since C is non-empty, pick any $c \in C$, and by coherence there is an instance $(x, y) \in f_g^{-1}(1) \times f_g^{-1}(0)$ as promised at a' (i. e., $x{\restriction}_{a'} \in P_{a'}^1$ and $y{\restriction}_{a'} \in P_{a'}^0$). Since $P_{a'}^0$ is disjoint from $P_{a'}^1$, $x{\restriction}_{a'} \neq y{\restriction}_{a'}$.
  - $a' \in A$ is alive: note that $|C{\restriction}_{a'}| \geq \textsc{AveDeg}_{a'}(C) \gg 1$, hence there are distinct $c^1 \neq c^0 \in C{\restriction}_{a'}$. Since $A \subseteq S$, $a' \in S$ is *same*, by coherence there is $x \in f_g^{-1}(1)$ with $x{\restriction}_{a'} = c^1$, and by coherence there is $y \in f_g^{-1}(0)$ with $y{\restriction}_{a'} = c^0$. Hence $x{\restriction}_{a'} \neq y{\restriction}_{a'}$.

In both cases, there is an instance $(x, y) \in f_g^{-1}(1) \times f_g^{-1}(0)$ such that $x\restriction_{a'} \neq y\restriction_{a'}$, thus the relevant entries of x and y are different at a (i. e., $x\restriction_{\delta^{\text{in}}(a)} \neq y\restriction_{\delta^{\text{in}}(a)}$). Then i cannot be a bit (position) relevant to both x and y, and Π cannot be output-relevant. $\square$

To conclude this subsection, the above adversary argument is connected with the Raz–McKenzie pebble game below.

Definition 4.17 (Initial Conditions)**.** At the root node r of the protocol Π for KW_f where $f := \mathsf{BDEP}_G^k$, $Y_r := f_r^{-1}(1) = f^{-1}(1)$, $N_r := f_r^{-1}(0) = f^{-1}(0)$, all vertices except the sink vertex τ are *same*, so $D_r := \{\tau\}$ and $S_r := V \setminus D_r$. Let $C_r := X\restriction_S$, $P_\tau^1 :=$ k-bit strings whose zeroth bit is 1 and $P_\tau^0 :=$ k-bit strings whose zeroth bit is 0.

Note that initially, all *same* vertices are alive ($A_r = S_r$), the data $\langle Y_r, N_r, S_r, D_r, C_r, \langle P_d^0, P_d^1\rangle_{d\in D}\rangle$ is coherent, and $\langle S_r, D_r, A_r\rangle$ is in a safe configuration. Throughout the protocol, dead vertices (i. e., non-alive vertices, $V \setminus A$) are precisely the vertices pebbled by *Pebbler* in the Raz–McKenzie pebble game, and the initial conditions correspond to the adversary making the first move of *Pebbler* to pebble w, and making the first (forced) move of *Colorer* to color τ red (i. e., $P = V \setminus A$ throughout, and the initial configuration is C_G^{RM} in RM_G, see Definition 3.15). Later in the protocol, when some vertex $a \in V$ loses too much entropy and dies, the adversary makes a move for *Pebbler* to pebble a, and then makes a move for *Colorer* under the optimal strategy against *Pebbler*, and (1) keeps a as *same* if *Colorer* colors a blue; or (2) marks a as *different* if *Colorer* colors a red. (Thus $B = S \setminus A$ throughout, see Definition 3.15.)

Claim 4.18 (Safe Till it is Over)**.** *Till the Raz–McKenzie pebble game is over, $\langle S, D, A\rangle$ remains a safe configuration.*

Proof. If $\langle S, D, A\rangle$ is not safe, then some *different* vertex $d \in D$ has all its in-neighbors *same* and dead, i. e., $\delta^{\text{in}}(d) \subseteq S \setminus A$. Since dead vertices are pebbled $V \setminus A = P$ and dead vertices are blue if *same* $B = S \setminus A$ (see Definition 3.15), it follows that $d \in P \setminus B$ is red while $\delta^{\text{in}}(d) \subseteq B$ are blue, so the Raz–McKenzie pebble game is over. $\square$

4.5 Recursive Lower Bound

This subsection formally proves Theorem 10, by enforcing the pebbling strategy in §4.4 with information theoretic (counting) arguments (Appendix A). Throughout this subsection, fix a directed acyclic graph $G = (V, E)$ whose Raz–McKenzie pebble game takes h time, and an output-relevant protocol Π for KW_f where $f := \mathsf{BDEP}_G^k$. For any real number $\alpha \geq 0$ and integer $0 \leq t \leq |V|$, consider the set of all Karchmer–Wigderson games $\mathsf{KW}_{Y,N}$ satisfying

- there is a node $g \in \Pi$ such that $Y \subseteq f_g^{-1}(1)$ and $N \subseteq f_g^{-1}(0)$, where the boolean function f is BDEP_G^k;
- there are sets $S \subseteq V$, $D = V \setminus S$, and for every $d \in D$, there are disjoint sets $P_d^0, P_d^1 \subset X\restriction_d$, such that $Y\restriction_d \subseteq P_d^1$ and $N\restriction_d \subseteq P_d^0$;
- there is a large set $C \subseteq X\restriction_A$ of values when restricted to a set $A \subseteq S$ of alive co-ordinates (under C, Definition 4.14), $|A| = t$, with at most α bits known about C, that is, $\alpha \geq \log_2(|X\restriction_A|/|C|) = tk - \log_2(|C|)$;
- C is thick, $\textsc{Thickness}(C) \geq K^{17/20}$ (Definition A.2 in Appendix A);
- C is common to both Y and N over A, in the sense that $C \subseteq (Y\restriction_A) \cap (N\restriction_A)$;

- the data $\big\langle Y, N, S, D, C, \langle P_d^0, P_d^1\rangle_{d\in D}\big\rangle$ is coherent at g; and
- the pebble configuration corresponding to $\langle S, D, A\rangle$ has value at least $t - |V| + h$, i.e., $\textsc{Val}(\langle\!\langle P, \bot, B\rangle\!\rangle) \geq t - |V| + h$ where $P := V \setminus A$ and $B := S \setminus A$ (Definitions 3.15 and 3.16).

Any such game $\mathsf{KW}_{Y,N}$ has data $\big\langle g, Y, N, S, D, C, \langle P_d^0, P_d^1\rangle_{d\in D}, A\big\rangle$, and denote its communication complexity under output-relevant protocols by $\mathrm{CC}^{\textsc{OutRel}} := \mathrm{CC}^{\textsc{OutRel}}(\mathsf{KW}_{Y,N})$. Note that $\langle S, D, A\rangle$ is in a safe configuration when $t > |V| - h + 1$ by Claim 4.18. In this case the data satisfy the conditions of Lemma 4.16. In addition to these data, the extra parameters α and t specify respectively the amount of information known about the common values C and the number of alive vertices $|A|$. Denote the collection of such games with parameters α and t as $\textsc{Games}\big[\alpha, t\big]$.

Definition 4.19 (Complexity Measure)**.** Let $\textsc{Comp}\big[\alpha, t\big]$ be the minimum communication complexity by output-relevant protocols solving any Karchmer–Wigderson game in $\textsc{Games}\big[\alpha, t\big]$. That is,

$$\textsc{Comp}\big[\alpha, t\big] := \min_{\mathsf{KW}_{Y,N} \in \textsc{Games}\left[\alpha, t\right]} \mathrm{CC}^{\textsc{OutRel}}(\mathsf{KW}_{Y,N}) \ .$$

The following lemma lower bounds the complexity measure, and follows the proof of the main theorem of Raz and McKenzie [RM99, §6].

Claim 4.20 (Recursive Lower Bound)**.** *When $K \geq |V|^{20}$ and $t > |V| - h + 1$,*

$$\textsc{Comp}\big[\alpha, t\big] \geq \min\Big\{\ \textsc{Comp}\big[\alpha + 2, t\big] + 1,\ \textsc{Comp}\big[\alpha - \tfrac{1}{20}k + 3, t - 1\big]\ \Big\} \ .$$

In particular,

$$\textsc{Comp}\big[\alpha, t\big] \geq \frac{1}{2}\Big[(t - |V| + h - 1)\big(\frac{k}{20} - 3\big) - \alpha\Big] \ .$$

Proof. Consider a Karchmer–Wigderson game $\mathsf{KW}_{Y,N}$ in $\textsc{Games}\big[\alpha, t\big]$ with data $\big\langle g, Y, N, S, D, C, \langle P_d^0, P_d^1\rangle_{d\in D}, A\big\rangle$. There are two cases:

1. for every $j \in A$, $\textsc{AveDeg}_j(C) \geq 8 \cdot K^{19/20}$, and
2. for some $j \in A$, $\textsc{AveDeg}_j(C) < 8 \cdot K^{19/20}$.

Then the first half of the lemma follows from Claims 4.21 and 4.22 below. Induction then gives the second half. ☐

Claim 4.21 (Recursive Lower Bound, Alive Case)**.** *Assume $t > |V| - h + 1$. If for every $j \in A$, we have $\textsc{AveDeg}_j(C) \geq 8 \cdot K^{19/20}$, then*

$$\mathrm{CC}^{\textsc{OutRel}} \geq \textsc{Comp}\big[\alpha + 2, t\big] + 1 \ .$$

Proof. Recall that Π denotes the output-relevant protocol solving the game. The node g cannot be an output node of Π by Lemma 4.16. Assume without loss of generality that Player 1 transmits the first bit at node $g \in \Pi$, which partitions Y into two sets Y_0 and Y_1 (respectively at nodes g_0 and g_1, children of g). Now, we have

$$|(Y_0{\restriction_A}) \cap (N{\restriction_A})| \geq |C|/2 \qquad \text{or} \qquad |(Y_1{\restriction_A}) \cap (N{\restriction_A})| \geq |C|/2 \ .$$

Assume the former without loss of generality, and let $C' := (Y_0{\restriction_A}) \cap (N{\restriction_A})$. The assumption on average degree, together with Lemma A.3, gives $\textsc{AveDeg}_j(C') \geq 4 \cdot K^{19/20}$ for every $j \in A$. Now Lemma A.6 gives a set $C'' \subseteq C'$ with $|C''| \geq |C'|/2 \geq |C|/4$ and $\textsc{Thickness}(C') \geq K^{17/20}$. Let $Y'' := \big\{x \in Y : x{\restriction_A} \in C''\big\}$ be the subset of Y consistent with C'' when restricted to A. Then $\mathsf{KW}_{Y'',N}$ is in $\textsc{Games}\big[\alpha + 2, t\big]$, and the lemma follows. (A bit more precisely, $\mathsf{KW}_{Y,N}$ is the same as $\mathsf{KW}_{Y'',N}$, except that g is updated to g_0, Y to Y'', and C to C''.) ☐

Claim 4.22 (Recursive Lower Bound, Dead Case)**.** *Assume that $K \geq |V|^{20}$. If for some $j \in A$, we have* $\textsc{AveDeg}_j(C) < 8 \cdot K^{19/20}$*, then*

$$\mathrm{CC}^{\textsc{OutRel}} \geq \textsc{Comp}\big[\alpha - \tfrac{1}{20}k + 3, t-1\big] \ .$$

Proof. We have $\textsc{AveDeg}_j(C) < 8 \cdot K^{19/20}$ and $\textsc{Thickness}(C) \geq K^{17/20}$. Let $A' := A \setminus \{j\}$ and $C' := C\restriction_{A'}$. Now Lemmas A.4 and A.5 give

$$\frac{|C'|}{|X\restriction_{A'}|} > \frac{|C|}{|X\restriction_{A}|}\frac{1}{8}K^{1/20} \qquad \text{and} \qquad \textsc{Thickness}(C') \geq K^{17/20} \ ,$$

hence $\log_2\big(|X\restriction_{A'}|/|C'|\big) < \alpha - \frac{1}{20}k + 3$. After making j dead, update $\langle S, D, A\rangle$ to $\langle S', D', A'\rangle$ so that $\textsc{Val}(\langle\!\langle P', \bot, B'\rangle\!\rangle) \geq \textsc{Val}(\langle\!\langle P, \bot, B\rangle\!\rangle) - 1$, where $P' = V \setminus A'$, $B' = S' \setminus A'$, $P = V \setminus A$, and $B = S \setminus A$ (Definitions 3.15 and 3.16). In case j is made *different*, we need two new sets P_j^0, P_j^1 of promised values at j, as given by Claim 4.23 (together with Y', N'). Otherwise, j is made *same*, and let $Y' := Y$ and $N' := N$. In either case, the game $\mathsf{KW}_{Y',N'}$ with data $\big\langle g, Y', N', S', D', C', \langle P_d^0, P_d^1\rangle_{d\in D'}, A'\big\rangle$ is in $\textsc{Games}\big[\alpha - \frac{1}{20}k + 3, t-1\big]$, and the lemma follows. ☐

Claim 4.23 (Symmetry Breaking)**.** *For $K \geq |V|^{20}$, if Y, N, A and C are such that $C \subseteq (Y\restriction_A) \cap (N\restriction_A)$, given $j \in A$ with $\textsc{MinDeg}_j(C) \geq K^{17/20}$, let $A' := A \setminus \{j\}$ and $C' := C\restriction_{A'}$, then there exist $Y' \subseteq Y$, $N' \subseteq N$, and disjoint $P_j^0, P_j^1 \subset X\restriction_j$, such that $C' \subseteq (Y'\restriction_{A'}) \cap (N'\restriction_{A'})$ and $Y'\restriction_j \subseteq P_j^1$ and $N'\restriction_j \subseteq P_j^0$.*

Proof. Randomly partition $X\restriction_j$ into P_j^1 and P_j^0, by including each string in P_j^1 independently with probability half, and let $P_j^0 := X\restriction_j \setminus P_j^1$. Let Y' be the subset of Y which when projected to j is in P_j^1, and similarly define N' from N and P_j^0. Now $C' \subseteq (Y'\restriction_{A'}) \cap (N'\restriction_{A'})$ fails to hold only when there is a $c' \in C'$ such that all extensions of c' are in P_j^1, or all are in P_j^0. Since $\textsc{MinDeg}_j(C) \geq K^{17/20}$, this happens with probability at most $|C'| \cdot 2^{-K^{17/20}+1} \leq K^{|V|} \cdot 2^{-K^{17/20}+1} \leq 2^{K^{1/20}\log_2 K - K^{17/20}+1} \ll 1$. Hence the claimed sets exist with overwhelming probability.[33] ☐

Theorem 10 (Lower Bound for Evaluation)**.** *For any directed acyclic graph G whose Raz–McKenzie pebble game takes h time, if $2^k \geq |V|^{20}$, then any output-relevant circuit computing BDEP_G^k has depth at least $(h-1)(\frac{k}{40} - 2) = \Omega(hk)$.*

5 Resolution Refutations

5.1 Size Lower Bound from Depth

For further background on resolution refutations of unsatisfiable formulas, see e. g., [Urq11, Nor12]. The empty, unsatisfiable formula is denoted as $\bot$.

Urquhart [Urq11] escalated the depth complexity of a resolution refutation to a size lower bound on tree-like resolution refutations, based on the Prover/Delayer game introduced by Pudlák–Impagliazzo [PI00] and employed by Ben-Sasson–Impagliazzo–Wigderson [BSIW04], with the substitution construction of Alekhnovich–Razborov [BS09] (denoted $\Sigma^{\oplus}$ below; for generalizations, see [BSN11]).

Lemma 5.1 (Size Lower Bound from Depth [Urq11, Theorem 5.4])**.** *If $\mathsf{Depth}(\Sigma \vdash \bot) \geq k$, then any tree-like resolution refutation of $\Sigma^{\oplus}$ has size at least 2^k.*

[33] Alternatively, the existence of the claimed set can be demonstrated by a deterministic greedy algorithm.

Based on Ben-Sasson–Wigderson [BSW01] which extends Raz–McKenzie [RM99], Urquhart then constructed a pebbling contradiction formula [Urq11, Theorem 4.6] by escalating the hardness of black pebble game [PTC76], separating the width and depth of resolution refutations. We will see that it suffices to escalate the hardness of reversible black pebble game, which turns out to be connected to the depth complexity of search problems.

Definition 5.2 (Pebbling Contradictions)**.** Let Σ_G denote the pebbling contradiction over G, which is a CNF boolean formula defined as follows. Σ_G has one boolean variable v for each vertex $v \in G$. Σ_G is the conjunction over the following clauses, and hence is unsatisfiable.

- for all source vertex v in G, Σ_G has a clause with a single positive literal v;
- for all non-source vertex v in G having in-neighbors $\delta^{\text{in}}(v)$, Σ_G has a clause $v \vee \bigvee_{u \in \delta^{\text{in}}(v)} \bar{u}$; and
- for the sink vertex τ of G, Σ_G has a clause with a single negative literal $\bar{\tau}$.

5.2 Tight Bounds for Tree-Like Resolution

For an unsatisfiable formula Σ, we will need the well-known isomorphism between (regular) tree-like resolution refutations for Σ and decision trees solving the search problem for Σ [BSIW04, Lemma 7].

Theorem 3 (Depth of Pebbling Contradictions)**.** *Fix a directed acyclic graph $G = (V, E)$ with a unique sink τ. The depth complexity of resolution refutation for Σ_G is* exactly *the pebble cost in the Raz–McKenzie pebble game to pebble the sink vertex of $\hat{G}$, where $\hat{G} := (V \cup \{\hat{\tau}\}, E \cup \{(\tau, \hat{\tau})\})$ is G augmented with an extra vertex $\hat{\tau}$ as the new sink.*

Proof. Concerning depth complexity, assume the resolution refutation is tree-like without loss of generality. Note that a minimum depth tree-like resolution must be regular (as pointed out by Urquhart [Urq11], this is proved by Grigori Tseitin [Tse70], or alternatively this follows from a simple tree pruning argument [Urq95]). Now it corresponds to a valid strategy in the Raz–McKenzie pebble game over $\hat{G}$. For the other direction, any valid strategy in the Raz–McKenzie pebble game over $\hat{G}$ clearly gives a (regular) tree-like resolution refutation for Σ_G. □

Theorem 4 (Tight Size Bounds for Tree-Like Resolution)**.** *The tree-like resolution refutation of $\Sigma_G^{\oplus}$ has size complexity $2^{\Theta(\textsc{Val}(G))}$.*

Proof. Since $\textsc{Val}(\hat{G}) \geq \textsc{Val}(G)$, Lemma 5.1 and Theorem 3 give the lower bound. For the upper bound, we show a tree-like resolution refutation of depth $O\big(\textsc{Val}(G)\big)$, using the fact that $\textsc{Val}(\hat{G}) \leq \textsc{Val}(G) + 1$. Note that a Raz–McKenzie strategy over $\hat{G}$ of value $\textsc{Val}(\hat{G})$ naturally gives a decision tree for $\Sigma_G^{\oplus}$ of depth $2\textsc{Val}(\hat{G})$, which in turn gives a resolution refutation of depth $2\textsc{Val}(\hat{G})$. □

Finally, we extend the lower bound on depth complexity to k-DNF-resolution refutations introduced by Krajíček [Kra01a]. For its motivation, see e. g., the survey by Nordström [Nor12]. We follow the standard to treat a term (i. e., a conjunction of literals) dually as a collection of literals.

Definition 5.3 (k-DNF-Resolution)**.** Lines in a k-DNF-resolution refutation are k-DNF formulas, derived using the following inference rules (A and B denote k-DNF formulas, S and T denote k-terms, and $l_1, \ldots, l_k$ denote literals):

$$\textbf{k-cut} \quad \frac{(l_1 \wedge \cdots \wedge l_{k'}) \vee A \quad \neg l_1 \vee \cdots \vee \neg l_{k'} \vee B}{A \vee B}, \text{ where } k' \le k.$$

$$\textbf{$\wedge$-introduction} \quad \frac{A \vee S \quad A \vee T}{A \vee (S \wedge T)}, \text{ where } |S \cup T| \le k.$$

$$\textbf{$\wedge$-elimination} \quad \frac{A \vee S}{A \vee T}, \text{ where } T \subseteq S.$$

$$\textbf{Weakening} \quad \frac{A}{A \vee B}, \text{ for any k-DNF formula } B.$$

Theorem 5. *Any k-DNF-resolution refutation of Σ_G has depth at least $1 + (\textsc{Val}(G) - 1)/k$.*

Proof. Imagine an adversary, who keeps track of a k-DNF formula A_d at 'depth' d in the refutation and a restriction ρ_d, satisfying the invariant that (1) ρ_d falsifies A_d; and (2) the configuration corresponding to ρ_d has value at least $\textsc{Val}(\hat{G}) - (d-1)k$, i. e., let B_d be the variables assigned $\textsc{True}$ under ρ_d, R_d $\textsc{False}$ (the extra sink of $\hat{G}$ is always assumed $\textsc{False}$), then $\textsc{Val}(\llparenthesis B_d, R_d \rrparenthesis) \ge \textsc{Val}(\hat{G}) - (d-1)k$.

The adversary starts with the unsatisfiable k-DNF formula $A_1 := \bot$ and $\rho_1 :=$ the unique sink of $\hat{G}$ is $\textsc{False}$, satisfying the invariant at 'depth' $d := 1$. If the adversary hits an axiom formula A_d from Σ_G, then ρ_d falsifies A_d, i. e., $\textsc{Val}(\llparenthesis B_d, R_d \rrparenthesis) = 1$, giving the required depth on the refutation.

Otherwise, we have A_d and ρ_d, where A_d is the result (i. e., on the bottom row) of an inference rule. We will locate A_{d+1} as one of the formulas on the top row of the inference rule, and update ρ_{d+1} appropriately. For the $\wedge$-elimination rule and the weakening rule, the adversary takes A_{d+1} as the only formula on the top row, and takes $\rho_{d+1} := \rho_d$ to maintain the invariant. For the $\wedge$-introduction rule, the adversary takes $\rho_{d+1} := \rho_d$, and takes A_{d+1} to be a formula on the top row that is falsified by ρ_{d+1}.

For the remaining, interesting case of a k-cut rule, the adversary maintains A_{d+1} and ρ_{d+1} by the recurrence of the Raz–McKenzie pebble game (Proposition 3.18). There are at most k fresh variables among the literals $l_1, \ldots, l_{k'}$ outside of the domain of ρ_d. At least one assignment to the fresh variables gives an extension ρ_{d+1} to ρ_d such that $\textsc{Val}(\llparenthesis B_{d+1}, R_{d+1} \rrparenthesis) \ge \textsc{Val}(\llparenthesis B_d, R_d \rrparenthesis) - k$, and at least one formula A_{d+1} on the top row is falsified by ρ_{d+1}. ☐

6 Some Related Approaches

We recall below some related approaches for separating complexity classes mostly around P.

Multi-Party Communication Complexity As an approach to separate ACC^0 from P, researchers considered the multi-player pointer jumping problem [Cha07,BC08,VW09], with the aim of proving a sufficiently strong lower bound in the number-on-forehead multi-party (simultaneous message) communication model. A variant of the problem with a tree structure, called tree pointer jumping problem [VW09], is like the tree evaluation problem with information flowing in the reverse direction (from root to leaves).

Extension to Karchmer–Wigderson framework Aaronson–Wigderson [AW09] extended the Karchmer–Wigderson framework [KW90] to consider a refereed communication game between two parties (verifiers) and an additional prover, where a sufficiently strong lower bound on communication complexity would separate NL from NP. Kol–Raz [KR13] extended the Aaronson–Wigderson framework of refereed communication game to a competing-prover protocol with two verifiers and two provers, and suggested it as an approach for separating NC from P.

Block-Respecting Simulations Lipton–Williams [LW12] recently suggested that a sufficiently strong lower bound on depth (e. g., $n^{1-O(1)}$) may be able to separate NC from P, even with a very weak lower bound on size (e. g., $n^{1+\Omega(1)}$), by using a block-respecting simulation to trade depth for size and non-uniformity. The idea of proving lower bounds by trading depth for size was due to Allender–Koucký [AK10].

Combinatorial Invariants Mulmuley–Sohoni [MS01, MS08] advocated the study of symmetry and invariants of the computational problems as an approach for separating VP from VNP, the non-uniform and algebraic analogue of P versus NP. One motivation is that Mulmuley [Mul99] applied semi-algebraic geometry to give a non-uniform and algebraic separation of $\mathsf{alg\text{-}NC}$ from $\mathsf{alg\text{-}P}$ and $\mathsf{alg\text{-}NC}^i$ from $\mathsf{alg\text{-}NC}^{i+1}$ on a restricted model of PRAM without bit operations, setting the stage for proving stronger lower bounds. Another motivation is that properties described by certain combinatorial invariants are unlikely to be large or natural in the sense of Razborov–Rudich [RR97], see e. g., Mulmuley [Mul11, §4.3].

Our Approach For comparison, our approach is closer to the competing prover protocols [KR13] than to the multi-party communication complexity approach [Cha07, BC08, VW09], due to the way that information is shared among the small number of parties involved (similar to [KW90, AW09]). Also, the study of the DAG evaluation problem (BDEP_G^k) or the Generation problem might provide the depth lower bounds required by block-respecting simulations [LW12] (recall the pebbling results in § 1.1). In terms of combinatorial invariants, instead of considering representation-theoretic, algebro-geometric invariants [MS01, MS08], we have been considering enumerative-combinatorial invariants shaped by pebbling strategies [Pot10, CP12] on monotone models. Our approach is inspired by the consideration of thrifty branching programs [CMW+12].

7 Future Directions

Problem 7.1. Is the Bennett–Dymond–Tompa–Raz–McKenzie pebble game PSPACE-complete?

Problem 7.1 and the conjecture of Urquhart on the complexity of the minimum depth of resolution refutations [Urq11, Problem 7.1] have been confirmed in an upcoming work of the author. Namely, it is PSPACE-complete to compute the pebble cost in the Bennett–Dymond–Tompa–Raz–McKenzie pebble game, and to compute the minimum depth of resolution refutations. The connection in Theorem 1 may explain why the one-player (irreversible) black pebble game is PSPACE-complete, while most PSPACE-complete games have two players.

Problem 7.2. Is it possible to connect other resources of the pebble games?

For example, this paper did not discuss the rounds in the Dymond–Tompa pebble game (or Raz–McKenzie pebble game), or the time in the reversible pebble game. It is of interest, since some resources of (the interpreted variant of) the Dymond–Tompa game [VT89] capture other computational resources, e. g., bounded alternations.

Problem 7.3. Would it help to prove lower bounds by considering the uniformity of the circuits?

It is not hard to see that BDEP_G^k is not solvable by AC^0 circuits when G is the pyramid graph of height $h = n^{\Theta(1)}$. Namely, when $k \approx \frac{1}{4}\log n$ and $h \approx n^{1/4}$, the average sensitivity of BDEP_G^k is $n^{\Theta(1)}$ (while any function computed by AC^0 circuits has average sensitivity $\log^{O(1)} n$ [LMN93]). It follows that BDEP_G^k is not computable by AC^0-uniform AC^0 circuits. Is it possible to relax the uniformity or the complexity of the circuits in this lower bound?

Problem 7.4. The Dymond–Tompa game lower bounds the scaling in complexity, for the problem of Generation on monotone switching networks, and for the problem of iterated indexing on output-relevant circuits, over any directed acyclic graph and for a wide range of parameters. To what extent, and on how general a model, does this correspondence hold?

The thrifty hypothesis of Cook, McKenzie, Wehr, Braverman, and Santhanam [CMW+12] can be rephrased as the conjecture that this correspondence holds for the black pebble game on the iterated indexing problem over the graph of binary trees, and on the model of branching programs, up to constant factors.

It would be interesting to refute or to establish the optimality of (the interpreted variant of) the Dymond–Tompa pebbling algorithms for space or parallel time: either (1) we get more space-efficient algorithms for graph reachability, or faster parallel speed-up for any P-complete problem (e. g., linear programming, semi-definite programming, circuit evaluation);[34] or (2) we get very strong complexity results, e. g., $\mathsf{L} \subset \mathsf{NL} \subset \mathsf{NC} \subset \mathsf{P}$ and $\mathsf{NC}^i \subset \mathsf{NC}^{i+1}$, and $\mathsf{DTime}[t] \not\subseteq \mathsf{ATime}\big[o(t/\log t)\big]$.

Acknowledgements

The author thanks James Cook, Stephen Cook, Yuval Filmus, Pierre McKenzie, Aaron Potechin, Robert Robere, and Dustin Wehr for their encouragements and their work on related research projects leading to this work. The author also benefited from discussions with Luca Trevisan and Ryan Williams, and from comments by Anand Bhaskar, Thomas Watson, and anonymous reviewers of CCC '2013.

References

[AB87] Noga Alon and Ravi B. Boppana, *The monotone circuit complexity of boolean functions*, Combinatorica **7** (1987), no. 1, 1–22.

[ABSRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson, *Space complexity in propositional calculus*, SIAM Journal on Computing **31** (2002), no. 4, 1184–1211.

[AD08] Albert Atserias and Víctor Dalmau, *A combinatorial characterization of resolution width*, Journal of Computer and System Sciences **74** (2008), no. 3, 323–334.

[AIK84] Akeo Adachi, Shigeki Iwata, and Takumi Kasai, *Some combinatorial game problems require $\Omega(n^k)$ time*, Journal of the ACM **31** (1984), no. 2, 361–376.

[AJPU07] Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart, *An exponential separation between regular and general resolution*, Theory of Computing **3** (2007), no. 1, 81–102.

[AK10] Eric Allender and Michal Koucký, *Amplifying lower bounds by means of self-reducibility*, Journal of the ACM **57** (2010), no. 3, 14:1–14:36.

[AW09] Scott Aaronson and Avi Wigderson, *Algebrization: A new barrier in complexity theory*, ACM Transactions on Computation Theory **1** (2009), no. 1, 2:1–2:54.

[BC08] Joshua Brody and Amit Chakrabarti, *Sublinear communication protocols for multi-party pointer jumping and a related lower bound*, 25th International Symposium on Theoretical Aspects of Computer Science (STACS 2008) (Dagstuhl, Germany) (Susanne Albers and Pascal Weil, eds.), Leibniz International Proceedings in Informatics

[34] Note that we consider $\mathsf{ATime}[\cdot]$ as parallel time, so some improvements [LV03, Wil05] do not apply.

(LIPIcs), vol. 1, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2008, pp. 145–165.

[BCGR92] Samuel R. Buss, Stephen A. Cook, Arvind Gupta, and Vijaya Ramachandran, *An optimal parallel algorithm for formula evaluation*, SIAM Journal on Computing **21** (1992), no. 4, 755–780.

[BEGJ98] Maria Luisa Bonet, Juan Luis Esteban, Nicola Galesi, and Jan Johannsen, *Exponential separations between restricted resolution and cutting planes proof systems*, FOCS (Palo Alto, California, USA), IEEE Computer Society, 1998, pp. 638–647.

[Ben73] Charles H. Bennett, *Logical reversibility of computation*, IBM Journal of Research and Development **17** (1973), no. 6, 525 –532.

[Ben89] ———, *Time/Space trade-offs for reversible computation*, SIAM Journal on Computing **18** (1989), no. 4, 766–776.

[Ber12] Christoph Berkholz, *On the complexity of finding narrow proofs*, FOCS, IEEE Computer Society, 2012, pp. 351–360.

[BHP10] Paul Beame, Trinh Huynh, and Toniann Pitassi, *Hardness amplification in proof complexity*, Proceedings of the 42nd ACM symposium on Theory of computing (New York, NY, USA), STOC '10, ACM, 2010, pp. 87–96.

[BIPS10] Paul Beame, Russell Impagliazzo, Toniann Pitassi, and Nathan Segerlind, *Formula caching in DPLL*, ACM Transactions on Computation Theory **1** (2010), no. 3, 9:1–9:33.

[BM91] David A. Mix Barrington and Pierre McKenzie, *Oracle branching programs and Logspace versus P*, Information and Computation **95** (1991), no. 1, 96–115.

[BOCIP02] Joshua Buresh-Oppenheim, Matthew Clegg, Russell Impagliazzo, and Toniann Pitassi, *Homogenization and the polynomial calculus*, Computational Complexity **11** (2002), no. 3, 91–108.

[BOGH$^+$06] Joshua Buresh-Oppenheim, Nicola Galesi, Shlomo Hoory, Avner Magen, and Toniann Pitassi, *Rank bounds and integrality gaps for cutting planes procedures*, Theory of Computing **2** (2006), no. 1, 65–90.

[BS90] Ravi B. Boppana and Michael Sipser, *The complexity of finite functions*, Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A), Elsevier and MIT Press, 1990, pp. 757–804.

[BS09] Eli Ben-Sasson, *Size-space tradeoffs for resolution*, SIAM Journal on Computing **38** (2009), no. 6, 2511–2525.

[BSIW04] Eli Ben-Sasson, Russell Impagliazzo, and Avi Wigderson, *Near optimal separation of tree-like and general resolution*, Combinatorica **24** (2004), no. 4, 585–603.

[BSN11] Eli Ben-Sasson and Jakob Nordström, *Understanding space in proof complexity: Separations and trade-offs via substitutions*, ICS (Bernard Chazelle, ed.), Tsinghua University Press, 2011, pp. 401–416.

[BSW01] Eli Ben-Sasson and Avi Wigderson, *Short proofs are narrow—resolution made simple*, Journal of the ACM **48** (2001), no. 2, 149–169.

[BTV01] Harry Buhrman, J. Tromp, and Paul Vitanyi, *Time and space bounds for reversible simulation*, arXiv:quant-ph/0101133 (2001), Journal of Physics A: Mathematical and General, 34(2001), 6821–6830.

[CEI96] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo, *Using the groebner basis algorithm to find proofs of unsatisfiability*, Proceedings of the twenty-eighth annual ACM symposium on Theory of computing (New York, NY, USA), STOC '96, ACM, 1996, pp. 174–183.

[Cha07] Amit Chakrabarti, *Lower bounds for multi-player pointer jumping*, IEEE Conference on Computational Complexity, IEEE Computer Society, 2007, pp. 33–45.

[Cha12] Siu On Chan, *Approximation resistance from pairwise independent subgroups*, Electronic Colloquium on Computational Complexity (ECCC) **19** (2012), 110.

[CKS81] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer, *Alternation*, Journal of the ACM **28** (1981), no. 1, 114133.

[CMW+12] Stephen A. Cook, Pierre McKenzie, Dustin Wehr, Mark Braverman, and Rahul Santhanam, *Pebbles and branching programs for tree evaluation*, ACM Transactions on Computation Theory **3** (2012), no. 2, 4:1–4:43.

[CP12] Siu Man Chan and Aaron Potechin, *Tight bounds for monotone switching networks via fourier analysis*, 2012, Journal version to appear in the Theory of Computing.

[CR79] Stephen A. Cook and Robert A. Reckhow, *The relative efficiency of propositional proof systems*, The Journal of Symbolic Logic **44** (1979), no. 1, 36–50.

[DLL62] Martin Davis, George Logemann, and Donald Loveland, *A machine program for theorem-proving*, Commun. ACM **5** (1962), no. 7, 394–397.

[DP60] Martin Davis and Hilary Putnam, *A computing procedure for quantification theory*, Journal of the ACM **7** (1960), no. 3, 201–215.

[DT85] Patrick W. Dymond and Martin Tompa, *Speedups of deterministic machines by synchronous parallel machines*, Journal of Computer and System Sciences **30** (1985), no. 2, 149 – 161.

[EGM04] Juan Luis Esteban, Nicola Galesi, and Jochen Messner, *On the complexity of resolution with bounded conjunctions*, Theoretical Computer Science **321** (2004), no. 2–3, 347–370.

[EIRS01] Jeff Edmonds, Russell Impagliazzo, Steven Rudich, and Jiri Sgall, *Communication complexity towards lower bounds on circuit depth*, Computational Complexity **10** (2001), 210–246, 10.1007/s00037-001-8195-x.

[EM10] Yara Elias and Pierre McKenzie, *DAG evaluation and the red-blue problem*, Advances and Applications of Automata on Words and Trees, 2010, http://www.dagstuhl.de/Materials/Files/10/10501/10501.McKenziePierre.Slides.pdf.

[ET01] Juan Luis Esteban and Jacobo Torán, *Space bounds for resolution*, Information and Computation **171** (2001), no. 1, 84–97.

[GH92] Mikael Goldmann and Johan Håstad, *A simple lower bound for monotone clique using a communication game*, Information Processing Letters **41** (1992), no. 4, 221 – 226.

[Gri91] Michelangelo Grigni, *Structure in monotone complexity*, Ph.D. thesis, Massachusetts Institute of Technology, June 1991.

[Gri01] Dima Grigoriev, *Linear lower bound on degrees of positivstellensatz calculus proofs for the parity*, Theoretical Computer Science **259** (2001), no. 1–2, 613–622.

[GS95] Michelangelo Grigni and Michael Sipser, *Monotone separation of logarithmic space from logarithmic depth*, Journal of Computer and System Sciences **50** (1995), no. 3, 433–437.

[Hak95] Armin Haken, *Counting bottlenecks to show monotone $P \neq NP$*, FOCS, IEEE Computer Society, 1995, pp. 36–40.

[HN12] Trinh Huynh and Jakob Nordström, *On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity*, STOC (Howard J. Karloff and Toniann Pitassi, eds.), ACM, 2012, pp. 233–248.

[HPV77] John Hopcroft, Wolfgang Paul, and Leslie Valiant, *On time versus space*, Journal of the ACM **24** (1977), no. 2, 332–337.

[HU07] Alexander Hertel and Alasdair Urquhart, *Game characterizations and the PSPACE-Completeness of tree resolution space*, Computer Science Logic (Jacques Duparc and Thomas Henzinger, eds.), Lecture Notes in Computer Science, vol. 4646, Springer Berlin / Heidelberg, 2007, pp. 527–541.

[HW97] Johan Håstad and Avi Wigderson, *Composition of the universal relation*, Advances in Computational Complexity Theory, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, 1997, pp. 119–134.

[JL74] Neil D. Jones and William T. Laaser, *Complete problems for deterministic polynomial time*, Proceedings of the Sixth Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC '74, ACM, 1974, pp. 40–46.

[JMNŽ12] Matti Järvisalo, Arie Matsliah, Jakob Nordström, and Stanislav Živný, *Relating proof complexity measures and practical hardness of SAT*, Principles and Practice of Constraint Programming (Michela Milano, ed.), Lecture Notes in Computer Science, Springer Berlin Heidelberg, jan 2012, pp. 316–331.

[Joh01] Jan Johannsen, *Depth lower bounds for monotone semi-unbounded fan-in circuits*, Theoretical Informatics and Applications **35** (2001), no. 3, 277–286.

[KAI79] Takumi Kasai, Akeo Adachi, and Shigeki Iwata, *Classes of pebble games and complete problems*, SIAM Journal on Computing **8** (1979), no. 4, 574–586.

[KPPY84] Maria Klawe, Wolfgang J. Paul, Nicholas Pippenger, and Mihalis Yannakakis, *On monotone formulae with restricted depth*, STOC (New York, NY, USA), ACM, 1984, pp. 480–487.

[KR13] Gillat Kol and Ran Raz, *Competing provers protocols for circuit evaluation*, Proceedings of the 4th conference on Innovations in Theoretical Computer Science (New York, NY, USA), ITCS '13, ACM, 2013, pp. 473–484.

[Kra01a] Jan Krajíček, *On the weak pigeonhole principle*, Fundamenta Mathematicae **170** (2001), no. 1, 123–140.

[Krá01b] Richard Král'ovič, *Time and space complexity of reversible pebbling*, SOFSEM 2001: Theory and Practice of Informatics (Leszek Pacholski and Peter Ružicka, eds.), Lecture Notes in Computer Science, vol. 2234, Springer Berlin / Heidelberg, 2001, pp. 292–303.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson, *Super-logarithmic depth lower bounds via the direct sum in communication complexity*, Computational Complexity **5** (1995), no. 3/4, 191–204.

[KW90] Mauricio Karchmer and Avi Wigderson, *Monotone circuits for connectivity require super-logarithmic depth*, SIAM Journal on Discrete Mathematics **3** (1990), no. 2, 255–265.

[Las01] Jean B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization **11** (2001), no. 3, 796–817.

[Lee59] Chang-Yeong Lee, *Representation of switching circuits by binary-decision programs*, Bell System Technical Journal **38** (1959), no. 4, 985–999.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan, *Constant depth circuits, fourier transform, and learnability*, Journal of the ACM **40** (1993), no. 3, 607–620.

[LMT00] Klaus-Jörn Lange, Pierre McKenzie, and Alain Tapp, *Reversible space equals deterministic space*, Journal of Computer and System Sciences **60** (2000), no. 2, 354 – 367.

[LNNW95] László Lovász, Moni Naor, Ilan Newman, and Avi Wigderson, *Search problems in the decision tree model*, SIAM Journal on Discrete Mathematics **8** (1995), no. 1, 119–132.

[LTV97] Ming Li, John Tromp, and Paul Vitanyi, *Reversible simulation of irreversible computation by pebble games*, arXiv:quant-ph/9703009 (1997), Physica D120 (1998) 168-176.

[LV96] Ming Li and Paul Vitanyi, *Reversibility and adiabatic computation: Trading time and space for energy*, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452** (1996), no. 1947, 769–789 (en).

[LV03] Richard Lipton and Anastasios Viglas, *Non-uniform depth of polynomial time and space simulations*, Fundamentals of Computation Theory (Andrzej Lingas and Bengt Nilsson, eds.), Lecture Notes in Computer Science, vol. 2751, Springer Berlin / Heidelberg, 2003, pp. 323–354.

[LW12] Richard J. Lipton and Ryan Williams, *Amplifying circuit lower bounds against polynomial time with applications*, IEEE Conference on Computational Complexity, 2012, pp. 1–9.

[McK10] Pierre McKenzie, 2010, Personal communication.

[MS01] Ketan D. Mulmuley and Milind Sohoni, *Geometric complexity theory I: An approach to the P vs. NP and related problems*, SIAM Journal on Computing **31** (2001), no. 2, 496–526.

[MS08] ______, *Geometric complexity theory II: towards explicit obstructions for embeddings among class varieties*, SIAM Journal on Computing **38** (2008), no. 3, 1175–1206.

[Mul99] Ketan D. Mulmuley, *Lower bounds in a parallel model without bit operations*, SIAM Journal on Computing **28** (1999), no. 4, 1460–1509.

[Mul11] ______, *On P vs. NP and geometric complexity theory*, Journal of the ACM **58** (2011), no. 2, 5:1–5:26.

[Nor12] Jakob Nordström, *Pebble games, proof complexity and time-space trade-offs*, To appear in Logical Methods in Computer Science.

[PH70] Michael S. Paterson and Carl E. Hewitt, *Comparative schematology*, Record of the Project MAC conference on concurrent systems and parallel computation (Jack B. Dennis, ed.), ACM, New York, NY, USA, 1970, pp. 119–127.

[PI00] Pavel Pudlák and Russell Impagliazzo, *A lower bound for DLL algorithms for k-sat*, Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms (Philadelphia, PA, USA), SODA '00, Society for Industrial and Applied Mathematics, 2000, pp. 128–136.

[Pot10] Aaron Potechin, *Bounds on monotone switching networks for directed connectivity*, 2010, An updated version to appear in Journal of the ACM.

[PTC76] Wolfgang J. Paul, Robert Endre Tarjan, and James R. Celoni, *Space bounds for a game on graphs*, Proceedings of the Eighth Annual ACM Symposium on Theory of Computing (New York, NY, USA), STOC '76, ACM, 1976, pp. 149–160.

[Raz85] Alexander A. Razborov, *Lower bounds on the monotone complexity of some boolean functions*, Soviet Mathematics Doklady **31** (1985), no. 2, 354–357.

[Raz91] ______, *Lower bounds for deterministic and nondeterministic branching programs*, FCT (Lothar Budach, ed.), Lecture Notes in Computer Science, vol. 529, Springer, 1991, pp. 47–60.

[Rei08] Omer Reingold, *Undirected connectivity in log-space*, Journal of the ACM **55** (2008), 17:1–17:24.

[RM99] Ran Raz and Pierre McKenzie, *Separation of the monotone NC hierarchy*, Combinatorica **19** (1999), no. 3, 403–435.

[RR97] Alexander A Razborov and Steven Rudich, *Natural proofs*, Journal of Computer and System Sciences **55** (1997), no. 1, 24–35.

[Ruz81] Walter Larry Ruzzo, *On uniform circuit complexity*, Journal of Computer and System Sciences **22** (1981), no. 3, 365–383.

[RW92] Ran Raz and Avi Wigderson, *Monotone circuits for matching require linear depth*, Journal of the ACM **39** (1992), no. 3, 736–744.

[Sav70] Walter J. Savitch, *Relationships between nondeterministic and deterministic tape complexities*, Journal of Computer and System Sciences **4** (1970), no. 2, 177–192.

[Sch08] Grant Schoenebeck, *Linear level lasserre lower bounds for certain k-CSPs*, IEEE, oct 2008, pp. 593–602.

[Set75] Ravi Sethi, *Complete register allocation problems*, SIAM Journal on Computing **4** (1975), no. 3, 226–248.

[Tse70] Grigori Samuilovich Tseitin, *On the complexity of derivation in propositional calculus*, Studies in Constructive Mathematics and Mathematical Logic, Part 2 (Anatol Oles'evich Slisenko, ed.), Consultants Bureau, New York, 1970, pp. 115–125.

[Tul09] Madhur Tulsiani, *CSP gaps and reductions in the lasserre hierarchy*, Proceedings of the 41st annual ACM symposium on Theory of computing (New York, NY, USA), STOC '09, ACM, 2009, pp. 303–312.

[Urq95] Alasdair Urquhart, *The complexity of propositional proofs*, The Bulletin of Symbolic Logic **1** (1995), no. 4, 425–467.

[Urq11] ______, *The depth of resolution proofs*, Studia Logica **99** (2011), no. 1, 349–364.

[VT89] H. Venkateswaran and Martin Tompa, *A new pebble game that characterizes parallel complexity classes*, SIAM Journal on Computing **18** (1989), no. 3, 533–549.

[VW09] Emanuele Viola and Avi Wigderson, *One-way multiparty communication lower bound for pointer jumping with applications*, Combinatorica **29** (2009), no. 6, 719–743.

[Weh11] Dustin Wehr, *Lower bound for deterministic semantic-incremental branching programs solving GEN*, CoRR **abs/1101.2705** (2011).

[Wil00] Ryan Williams, *Space-efficient reversible simulations*, Tech. report, 2000.

[Wil05] ______, *Parallelizing time with polynomial circuits*, Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures (New York, NY, USA), SPAA '05, ACM, 2005, p. 171175.

A Bounds on Information

This appendix collects the information theoretic (counting) arguments of Raz and McKenzie [RM99] used in this work. It may be a good idea to consult [RM99] (and also [EIRS01] that inspired [RM99], e. g., on the notion of predictability) for the intuition behind the information theoretic arguments (used in the depth lower bounds in restricted models [RM99, BEGJ98, Joh01, EM10]).

Let $X := [K]^\ell$ be an ℓ-fold product space and let C be a subset of X. Given a co-ordinate $j \in [\ell]$, define the bipartite graph $\textsc{Graph}_j(C) := \langle V_L, V_R, E\rangle$, where $V_L := C{\restriction}_j$ and $V_R := C{\restriction}_{[\ell]\setminus j}$ and $(v_L, v_R) \in E$ iff there is a $c \in C$ such that $c{\restriction}_j = v_L \in V_L$ and $c{\restriction}_{[\ell]\setminus j} = v_R \in V_R$.

Definition A.1 (Average Degree [RM99])**.** Given $j \in [\ell]$, we have $\textsc{Graph}_j(C) = \langle V_L, V_R, E\rangle$ and

$$\textsc{AveDeg}_j(C) := \frac{|E|}{|V_R|} = \frac{|C|}{|C{\restriction}_{[\ell]\setminus j}|} \ .$$

Definition A.2 (Min Degree and Thickness [RM99]). Given $j \in [\ell]$, we have $\textsc{Graph}_j(C) = \langle V_L, V_R, E\rangle$ and

$$\textsc{MinDeg}_j(C) := \min_{v_R \in V_R} \deg(v_R) \ .$$

Note that $\textsc{MinDeg}_j(C) > 0$, by definition of projection. Now

$$\textsc{Thickness}(C) := \min_{j\in[\ell]} \textsc{MinDeg}_j(C) \ .$$

Lemma A.3 (Large Size means Large Average Degree [RM99, Claim 5.1], [EIRS01, Lemma 4]). *Let $C' \subseteq C$. Then for any j,*

$$\textsc{AveDeg}_j(C') \geq \frac{|C'|}{|C|} \cdot \textsc{AveDeg}_j(C) \ .$$

Lemma A.4 (Entropy Refill). *For any $j \in [\ell]$,*

$$\frac{|C{\restriction}_{[\ell]\setminus j}|}{K^{t-1}} = \frac{|C|}{K^t}\frac{K}{\textsc{AveDeg}_j(C)} \ .$$

Lemma A.5 (Dropping Index does not Drop Thickness [RM99, Claim 5.2]). *For any $j \in [\ell]$,*

$$\textsc{Thickness}(C{\restriction}_{[\ell]\setminus j}) \geq \textsc{Thickness}(C) \ .$$

Lemma A.6 (Distilling Thickness from Average Degree [RM99, Corollary 5.4]). *Assume that $K \geq \ell^{20}$. If for every j, $\textsc{AveDeg}_j(C) \geq 4 \cdot K^{19/20}$, then there exists $C' \subseteq C$ such that:*

1. $|C'| \geq |C|/2$, *and*
2. $\textsc{Thickness}(C') \geq K^{17/20}$.

B Nondeterministic DAG Evaluation Problem

This appendix specializes the communication framework of Dart game to the Generation problem over a directed acyclic graph G, denoted Dart^k_G. As a result, this section generalizes the lower bounds on monotone circuit depth for Generation by Raz–McKenzie [RM99] to any directed acyclic graph. This has been claimed by Elias–McKenzie [EM10], which do not seem to be published. Results in this appendix are not original, but are included for ease of reference.

Appendix B.1 recalls the hardness escalation argument of Raz–McKenzie [RM99] for translating the decision tree complexity of the Raz–McKenzie pebble game to monotone communication complexity, and Appendix B.2 proves a lower bound on monotone circuit depth as Theorem 13.

Pierre McKenzie [McK10] introduced the following computational problem, which is roughly a monotone variant of the DAG Evaluation Problem, and lies in the combinatorial core of the lower bound on monotone circuit depth for Generation [RM99, Joh01, EM10]. This problem can be seen as a parameterized version of the P-complete problem of monotone circuit evaluation. By studying a slice of the problem (for a fixed graph G and constant k), we can focus on the combinatorics of the 'flow of values' over the graph. Indeed, we will analyze it to give a lower bound on the depth of monotone circuits solving the problem of Generation, whose $\textsc{Yes}$-instances have a structure of G.

Definition B.1 (Nondeterministic DAG Evaluation Problem over G)**.** Consider a DAG G and a bit-length parameter $k \in \mathbb{N}$. Denote the set of k-bit strings as $\{0,1\}^k \cong [K]$, where $K := 2^k$. The Nondeterministic DAG Evaluation Problem over G (NDEP_G^k) is specified by the following.

Input For every vertex $a \in V$, there is a function $t_a \colon [K]^{\delta^{\text{in}}(a)} \to 2^{[K]}$.[35] The input to NDEP_G^k enumerates the n bits of $\langle t_a \rangle_{a \in V}$ as n boolean variables where $n := K \sum_{a \in V} K^{\deg^{\text{in}}(a)}$.

'Computation' Define inductively the sets of values $\langle z_a \rangle_{a \in V} \in (2^{[K]})^V$ by

$$z_a := \bigcup_{v\restriction_{\delta^{\text{in}}(a)} \in z\restriction_{\delta^{\text{in}}(a)}} t_a\big(v\restriction_{\delta^{\text{in}}(a)}\big) \in 2^{[K]}$$

for $a \in V$.[36] That is, the set z_a is union of t_a applied to the sets at the in-neighbors of a.[35]

Output The output of NDEP_G^k is the tuple of sets $\langle z_w \rangle_{w \in W} \in (2^{[K]})^W$.

Henceforth, without loss of generality, focus on DAGs with exactly one sink vertex $\mathrm{\tau}$. The interest is in the boolean circuit depth complexity of computing a decision version of NDEP_G^k (as opposed to a $2^{[K]}$-valued function).

Definition B.2 (Boolean Nondeterministic DAG Evaluation Problem)**.** Fix a non-constant monotone boolean function τ on subsets of k-bit strings $\tau \colon 2^{\{0,1\}^k} \to \{0,1\}$, say the dictatorship $\tau(S) := (S \ni 0^k)$ for $S \subseteq \{0,1\}^k$.[37] The Boolean Nondeterministic DAG Evaluation Problem (BNDEP_G^k) seeks to compute $\tau(z_{\mathrm{\tau}})$.

B.1 Dart Game over G: Structured Protocol for Hardness Escalation

To study the depth complexity of monotone circuits, Raz–McKenzie introduced the Dart game framework [RM99, §2] based on the communication game of Karchmer–Wigderson [KW90]. Raz–McKenzie then specialized the Dart game to the Generation problem with a structure of a pyramid graph [RM99, §3] or of a line graph [RM99, §4.1]. Central to the lower bound argument in [RM99] is the Raz–McKenzie pebble game [McK10, EM10]. McKenzie [McK10] and Elias–McKenzie [EM10] observed that the lower bounds on monotone circuit depth in [RM99] can be extended to the Generation problem with a structure of any directed acyclic graph G, and the lower bounds scale in the same way as the value of the Raz–McKenzie pebble game over G. In order words, the Raz–McKenzie pebble game abstracts away the graph structure from the lower bound argument of [RM99].

Definitions B.4 and B.5 specialize notions from the Dart game of Raz–McKenzie [RM99, §2] to the Generation problem over directed acyclic graphs, denoted Dart_G^k. Lemma B.6 (based on the adversarial strategy of Raz–McKenzie [RM99, §3.3 and §4.1]) shows that structured protocols for this specialized Dart game is effectively playing the Raz–McKenzie pebble game.

Notation B.3 (Index and Table)**.** Consider a DAG $G = (V, E)$ and a bit-length parameter $k \in \mathbb{N}$. Denote $K := 2^k$. Let $\tilde{V} := V \setminus \{\mathrm{\tau}\}$ be the non-sink vertices. Let $X := [K]^{\tilde{V}}$ be indices, and $Y := (\{0,1\}^{[K]})^{\tilde{V}}$ be tables, over $\tilde{V}$. $x\restriction_a \in [K]$ denotes the entry of $x \in X$ indexed by $a \in \tilde{V}$, and $y\restriction_a := \big\{ i \in [K] \colon (i, a) \in y \big\}$ denotes the restriction of $y \in Y$ to $a \in \tilde{V}$.[38]

[35] Note that for a source vertex $a \in U$, its function t_a degenerates to have a domain of $[K]^{\emptyset}$, hence the function $t_a \in 2^{[K]}$ can be treated as a subset of k-bit string. Thus its set z_a is just its function $t_a \in [K]$ treated as a subset of k bit-string.

[36] Note that $z\restriction_{\delta^{\text{in}}(a)} \in (2^{[K]})^{\delta^{\text{in}}(a)}$ is identified with a (product) subset $z\restriction_{\delta^{\text{in}}(a)} \subseteq [K]^{\delta^{\text{in}}(a)}$, by $v\restriction_{\delta^{\text{in}}(a)} \in z\restriction_{\delta^{\text{in}}(a)}$ iff $v\restriction_{a'} \in z\restriction_{a'}$ for all $a' \in \delta^{\text{in}}(a)$.

[37] All non-constant monotone boolean functions are equivalent with respect to the (restricted) lower bounds in this work. The dictatorship is chosen here since its computation is trivial, i. e., takes no extra depth.

[38] A subset $S \subseteq [K]$ is identified with its indicator function $\chi_S \in \{0,1\}^{[K]}$, where $\chi_S(i) = 1$ iff $i \in S$.

Definition B.4 (Dart Game over G)**.** The Dart Game over G (Dart_G^k) is a (co-operative) communication game played between two parties, the Index Party and the Table Party. The Index Party is given the indices over vertices $x \in X := [K]^{\tilde{V}}$, and the Table Party is given the values of the tables $y \in Y := (\{0,1\}^{[K]})^{\tilde{V}}$. For each input instance $(x, y) \in X \times Y$ to Dart_G^k, define the boolean values $u_a := (x{\restriction_a} \in y{\restriction_a})$ for $a \in \tilde{V}$, Define $u_w := \textsc{False}$ for $w \in W$. They communicate to locate $a \in V$ such that $u_a = \textsc{False}$ but $u_{a'} = \textsc{True}$ for all $a' \in \delta^{\text{in}}(a)$. Denote the communication complexity of Dart_G^k by $\mathrm{CC}(\mathsf{Dart}_G^k)$.

Definition B.5 (Structured Communication Protocol [RM99, §2.2])**.** A communication protocol for Dart_G^k is *structured* if, in each round, the Index Party sends the index $x{\restriction_a}$ for some $a \in \tilde{V}$ (using k bits),[39] and the Table Party sends the value $u_a = (x{\restriction_a} \in y{\restriction_a})$ (using one bit), and they repeat the above until they have located the answer $a \in V$. The *structured complexity* of G (denoted as $\mathrm{SC}(G)$) is the number of rounds used by the shortest structured protocol to solve Dart_G^k.

Theorem 11 (Structured Complexity [RM99, Theorem 2.1])**.** *If $2^k \geq |V|^{20}$, then $\mathrm{CC}(\mathsf{Dart}_G^k) = \mathrm{SC}(G)\Omega(k)$, where the constant in $\Omega(\cdot)$ is independent of G.*

Lemma B.6 (Structured Protocol and Raz–McKenzie Pebble Game)**.** $\mathrm{SC}(G) = h - 1$ *iff RM_G takes h time to play.*

Proof. Consider a round-by-round isomorphism between (the subgames of) structured Dart_G^k and RM_G as follows. Namely, a certain round of an execution of a structured protocol of Dart_G^k is identified with the configuration $\langle\!\langle P, \perp, B \rangle\!\rangle$ in RM_G (see Definition 3.15), where $P \subseteq V$ are the vertices whose values are known (including $\tau \in W \subseteq V$), and $B \subset P$ are the vertices whose values are $\textsc{True}$ in Dart_G^k. Under this mapping, structured Dart_G^k and RM_G have the same initial configuration, and Dart_G^k is over iff RM_G is over. Also, a round identified with $\langle\!\langle P, \perp, B \rangle\!\rangle$ is followed by a next round identified with $\langle\!\langle P', \perp, B' \rangle\!\rangle$ iff $\langle\!\langle P, \perp, B \rangle\!\rangle \vdash C \vdash \langle\!\langle P', \perp, B' \rangle\!\rangle$ for some *Colorer* configuration C in $\mathsf{RM}_{\tilde{G}}$. Finally, the value of a round (i. e., the number of additional rounds before Dart_G^k is over) in structured Dart_G^k agrees with the value of $\langle\!\langle P, \perp, B \rangle\!\rangle$ in $\mathsf{RM}_{\tilde{G}}$ (see Definition 3.16), due to the same recursive dependence: the Index Party (*Pebbler*) wants to choose a vertex $a \in \tilde{V}$, so that no matter the Table Party (*Colorer*) replies with u_a is $\textsc{True}$ (blue) or $\textsc{False}$ (red), the next round has a value smaller by at least one. □

B.2 Lower Bound for Nondeterministic Evaluation

Recall that given a monotone boolean function $f\colon \{0,1\}^n \to \{0,1\}$, the monotone Karchmer–Wigderson game ($\mathsf{m\text{-}KW}_f$) is a communication game between two parties defined as follows: Party 1 (the $\textsc{Yes}$ party) is given a $\textsc{Yes}$ instance $x \in f^{-1}(1)$, Party 0 (the $\textsc{No}$ party) is given a $\textsc{No}$ instance $y \in f^{-1}(0)$, and they communicate to locate a bit position $i \in [n]$ where $x_i > y_i$. Karchmer and Wigderson observed that the monotone communication complexity captures exactly the monotone circuit depth.[31]

Theorem 12 (Karchmer–Wigderson [KW90, Theorem 2.2][31])**.** *The depth complexity of f on monotone boolean circuits is exactly the communication complexity of $\mathsf{m\text{-}KW}_f$.*

Lemma B.9 naturally generalizes [RM99, Lemma 3.5 and §4.1] (from the pyramid graphs and the line graphs respectively) to any directed acyclic graph.

[39] The two parties should agree on a common protocol before communicating to specify the vertex $a \in \tilde{V}$ in each round, so the vertex $a \in \tilde{V}$ is understood to both parties and need not be communicated.

Notation B.7 (Variables for BNDEP_G^k)**.** Recall that an instance to BNDEP_G^k specifies functions $t_a\colon [K]^{\delta^{\text{in}}(a)} \to 2^{[K]}$ for all $a \in V$, encoded as $n := K \sum_{a\in V} K^{\deg^{\text{in}}(a)}$ bits. Hence any tuple of values $v{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)} \in [K]^{\{a\}\cup\delta^{\text{in}}(a)}$ indexes an input bit (variable) of t_a, namely, whether $t_a\big(v{\restriction}_{\delta^{\text{in}}(a)}\big) \subseteq [K]$ contains $v{\restriction}_a \in [K]$.

Definition B.8 (Index and Table Instances)**.** X is identified with a subset of Yes instances to $f := \mathsf{BNDEP}_G^k$, Y a subset of No instances, as follows. Given any $x \in X = [K]^{\tilde{V}}$, let its Yes-extension $\hat{x} \in [K]^V$ be such that $\hat{x}{\restriction}_{\tilde{V}} = x$ and $\hat{x}{\restriction}_\tau = 0^k \in \tau^{-1}(1) \subset [K]$ (Definition B.2). Consider a Yes instance $I(x) \in f^{-1}(1)$ where a boolean variable indexed by $v{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)}$ is True in $I(x)$ iff $v{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)} = \hat{x}{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)}$. Note that $v_a \in z_a$ in $I(x)$ iff $v_a = \hat{x}_a$ (by induction on $a \in V$). Given any $y \in Y = (2^{[K]})^{\tilde{V}}$, let its No-extension $\hat{y} \in (2^{[K]})^V$ be such that $\hat{y}{\restriction}_{\tilde{V}} = y$ and $\hat{y}{\restriction}_w = \emptyset \subset \tau^{-1}(0) \subset [K]$ (Definition B.2). Consider a No instance $I(y) \in f^{-1}(0)$ where a boolean variable indexed by $v{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)}$ is False in $I(y)$ iff (i) $v{\restriction}_{a'} \in \hat{y}{\restriction}_{a'}$ for all $a' \in \delta^{\text{in}}(a)$; and (ii) $v{\restriction}_a \notin \hat{y}{\restriction}_a$. Note that if $v_a \in z_a$ in $I(y)$, then $v_a \in \hat{y}_a$ (by induction on $a \in V$).

Lemma B.9 (Reduction in Communication Games)**.** $\mathrm{CC}(\mathit{Dart}_G^k) \leq \mathrm{CC}(\textit{m-KW}_{\mathit{BNDEP}_G^k})$.

Proof. Given a communication protocol Π for $\mathsf{m\text{-}KW}_f$ where $f := \mathsf{BNDEP}_G^k$, we construct a communication protocol Π' for Dart_G^k of the same complexity. For any input $(x', y') \in X \times Y$ to Dart_G^k, the two parties can construct an input $(x, y) := \big(I(x'), I(y')\big) \in f^{-1}(1) \times f^{-1}(0)$ to $\mathsf{m\text{-}KW}_f$ without communicating. Now they run the protocol Π for $\mathsf{m\text{-}KW}_f$, getting a bit position $i \in [n]$ such that $x_i > y_i$, corresponding to a boolean variable indexed by $v{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)} \in [K]^{\{a\}\cup\delta^{\text{in}}(a)}$. Since the boolean variable is True in $x = I(x')$, we have $v{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)} = \hat{x}'{\restriction}_{\{a\}\cup\delta^{\text{in}}(a)}$. And as the boolean variable is False in $y = I(y')$, we have (i) $v{\restriction}_{a'} \in \hat{y}'{\restriction}_{a'}$ for all $a' \in \delta^{\text{in}}(a)$; and (ii) $v{\restriction}_a \notin \hat{y}'{\restriction}_a$. It follows that u_a is False but $u_{a'} = $ True for all $a' \in \delta^{\text{in}}(a)$ (Definition B.4). Hence they can output a for Dart_G^k, and this completes the description of Π'. □

Theorem 13 is a corollary of Lemmas B.6 and B.9 and Theorems 11 and 12. It implies a lower bound on the depth of monotone circuits solving the Generation problem whose Yes-instances have a structure of G.

Theorem 13 (Lower Bound for Nondeterministic Evaluation)**.** *For any directed acyclic graph G whose Raz–McKenzie pebble game takes h time, if $2^k \geq |V|^{20}$, then any monotone circuit computing BNDEP_G^k has depth at least $\Omega(hk)$, where the constant in $\Omega(\cdot)$ is independent of G.*

ECCC ISSN 1433-8092
http://eccc.hpi-web.de