PROOF COMPLEXITY AND THE BINARY ENCODING OF COMBINATORIAL PRINCIPLES*

STEFAN DANTCHEV †, NICOLA GALESI ‡, ABDUL GHANI †, AND BARNABY MARTIN †

Key words. Propositional proof complexity, Resolution, Lift-and-Project Methods, Sherali-Adams, Binary encoding

AMS subject classifications. 68Q25, 03F20

Abstract. We consider proof complexity in light of the unusual *binary* encoding of certain combinatorial principles. We contrast this proof complexity with the normal unary encoding in several refutation systems, based on Resolution and Sherali-Adams.

We firstly consider Res(s), which is an extension of Resolution working on s-DNFs. We prove an exponential lower bound of $n^{\Omega(k)/\mathrm{d}(s)}$ for the size of refutations of the binary version of the k-Clique Principle in Res(s), where $s = o((\log\log n)^{1/3})$ and $\mathrm{d}(s)$ is a doubly exponential function. Our result improves that of Lauria et al. who proved a similar lower bound for Res(1), i.e. Resolution. For the k-Clique and other principles we study, we show how lower bounds in Resolution for the unary version follow from lower bounds in Res($\log n$) for the binary version, so we start a systematic study of the complexity of proofs in Resolution-based systems for families of contradictions given in the binary encoding.

We go on to consider the binary version of the (weak) Pigeonhole Principle Bin-PHP^m_n. We prove that for any $\delta, \epsilon > 0$, Bin-PHP^m_n requires refutations of size $2^{n^{1-\delta}}$ in Res(s) for $s = O(\log^{\frac{1}{2}-\epsilon} n)$. Our lower bound cannot be improved substantially with the same method since for $m \geq 2^{\sqrt{n \log n}}$ we can prove there are $2^{O(\sqrt{n \log n})}$ size refutations of Bin-PHP^m_n in Res($\log n$). This is a consequence of the same upper bound for the unary weak Pigeonhole Principle of Buss and Pitassi.

We contrast unary versus binary encoding in the Sherali-Adams (SA) refutation system where we prove lower bounds for both rank and size. For the unary encoding of the Pigeonhole Principle and the Ordering Principle, it is known that linear rank is required for refutations in SA, although both admit refutations of polynomial size. We prove that the binary encoding of the (weak) Pigeonhole Principle Bin-PHP^m_n requires exponentially-sized (in n) SA refutations, whereas the binary encoding of the Ordering Principle admits logarithmic rank, polynomially-sized SA refutations.

We continue by considering a natural refutation system we call "SA+Squares", intermediate between SA and Lasserre (Sum-of-Squares). This has been studied under the name static-LS$^\infty_+$ by Grigoriev et al. In this system, the unary encoding of the Linear Ordering Principle LOP$_n$ requires $O(n)$ rank while the unary encoding of the Pigeonhole Principle becomes constant rank. Since Potechin has shown that the rank of LOP$_n$ in Lasserre is $O(\sqrt{n}\log n)$, we uncover an almost quadratic separation between SA+Squares and Lasserre in terms of rank. Grigoriev et al. noted that the unary Pigeonhole Principle has rank 2 in SA+Squares and therefore polynomial size. Since we show the same applies to the binary Bin-PHP$^{n+1}_n$, we deduce an exponential separation for size between SA and SA+Squares.

1. Introduction.

Various fundamental combinatorial principles used in proof complexity may be given in first-order logic as sentences φ with no finite models. Riis discusses in [64] how to generate from φ a family of CNFs $\{\varphi_n\}_{n\in\mathbb{N}}$, such that φ_n encodes the fact that φ has a model of size n. If φ has no finite models, this family $\{\varphi_n\}_{n\in\mathbb{N}}$ will be of unsatisfiable CNFs. Following Riis, it is typical to encode the

*This paper is an expanded version of "Resolution and the binary encoding of combinatorial principles" from the 34th Computational Complexity Conference (CCC) 2019 and "Sherali-Adams and the binary encoding of combinatorial principles" from the 14th Latin American Theoretical Informatics Symposium (LATIN) 2020.

†Department of Computer Science, Durham University, U.K.

‡Dipartimento di Ingegneria Informatica, Automatica e Gestionale "A. Ruberti", Sapienza Università Roma.

existence of such model with a big disjunction of the form $v_{\mathbf{a},1} \vee \ldots \vee v_{\mathbf{a},n}$,[**] that we designate the *unary encoding*. As can be observed, in the unary encoding to capture the existence of a model one uses as many literals as elements of the model's domain. However one can think of encoding the existence of such a model *succinctly* by using a *binary encoding*: each element j of the model can be captured by specifying $\log n$ bits, and then using variables $\omega^{j_h}_{\mathbf{a},h}$ capturing the parity j_h of each bit h of the binary encoding $\mathrm{bin}(j)$ of j. The binary encoding of combinatorial statements is a natural extension to propositional formulas of the notion of the *bit-graph representation* of functions. As a simple example of the binary encoding, consider to have a disjunction of 4 variables $w_0 \vee w_1 \vee w_2 \vee w_3$. We can encode this in binary using two variables ω_0 and ω_1, where we express w_0 as $\neg\omega_0 \wedge \neg\omega_1$, w_1 as $\neg\omega_0 \wedge \omega_1$, w_3 as $\omega_0 \wedge \neg\omega_1$, and w_4 as $\omega_0 \wedge \omega_1$.

One of the main aims of proof complexity is to find hard combinatorial properties whose propositional translation might lead to hard-to-prove formulas. The complexity of proving formulas in proof systems is measured as a function of the size (or other measures like, for instance, the maximal width in CNFs) of the formula to be proved. Combinatorial principles encoded in binary are interesting to study in proof complexity: on the one hand they preserve the combinatorial structure of the principle encoded, and on the other hand they give a more succinct propositional representation of the formula to be studied that could make easier the task of obtaining strong lower bounds. In fact in many recent works the binary encoding of combinatorial principles were used to prove hardness results for the complexity of proofs in several distinct proof systems and for different proof complexity measures.

In light of this, Thapen and Skelley considered in [68] the binary encoding of a combinatorial principle on k-turn games GI_3 and proved an exponential lower bound for refuting GI_3 in *Resolution*. Several other examples followed and more recently the binary encoding of the *Pigeonhole principle* has been considered in several works. In [41], it was used to prove new size lower bounds for Cutting Planes, by a new technique. In [11], it was used to prove lower bounds for $\mathrm{Res}(s)$ refutations (which involved the relativised version of the weak pigeonhole principle). Very recently in [32], it is used for the generalisation and simplification of the NP-completeness of automatising Resolution [10]. Finally, in another recent work [42], where it is called the *bit Pigeonhole Principle*, it is used in a proof of lower bounds for k-party communication complexity. However, binary encodings are meaningful to apply also to other combinatorial principles as well and also to other proof complexity measures. The work [51] solves an important open problem on the complexity of proofs in Resolution of a combinatorial principle expressing the presence of a k-clique in graphs, in the case of a binary encoding. Several techniques to prove *space* proof complexity lower bounds were applied successfully on the binary encoding of principles [34, 21, 22].

In all these cases, considering the binary encoding led to significant lower bounds in an easier way than for the unary case. Of course the idea of considering succinct encodings is not new and is not limited to proof complexity. Use of the binary encoding in bounded arithmetic seems to predate its use in proof complexity. Furthermore, since the succinctness of the encoding of the formulas might affect the running time of routines having formulas as input, it is no surprise that binary encodings have been studied systematically in the "dual" applied area of SAT-solving [47, 56], where it is

[**]Here $\mathbf{a}$ is the sequence of universal variables preceding some single existential variable the disjunction is witnessing. Such a disjunction appears for all existential variables in φ. An example of this translation of a first-order sentence appears at the opening of Section 7.

usual to try different encodings of the 1-from-n constraint to speed-up the running time of SAT-solvers both on satisfiable and unsatisfiable formulas. In [56, 70], what we call the binary encoding is referred to as *logarithmic*.

Merging the results in [27, 28], the central thrust of this work is to start a systematic study contrasting the proof complexity between the unary and binary encodings of natural combinatorial principles. To compare the complexity of proving propositional binary and unary encodings we will consider several refutation systems, three distinct combinatorial principles (and their variants) and different complexity measures. One of our main contributions is a lower bound similar to that obtained in [51] for the binary principle expressing the presence of k-cliques in graphs, for an extension of the Resolution system which allows bounded conjunctions, $\mathrm{Res}(s)$. In obtaining this lower bound we devise a new technique to prove size lower bounds in $\mathrm{Res}(s)$ which is suitable for binary encodings and which we also successfully apply to the case of the Pigeonhole Principle.

2. Overview of the results.

We consider three main combinatorial principles to contrast binary and unary proof complexity: (1) the k-Clique Formulas, $\mathrm{Clique}_k^n(G)$; (2) the (weak) Pigeonhole Principle PHP_n^m; and (3) the (Linear) Ordering Principle, $\mathrm{(L)OP}_n$.

The *k-Clique Formulas* introduced in [17, 18, 13] are formulas stating that a given graph G does have a k-clique and are therefore unsatisfiable when G does not contain a k-clique. The Pigeonhole Principle states that a total mapping $f : [m] \to [n]$ has necessarily a collision when $m > n$. Its propositional formulation in the negation, PHP_n^m is well-studied in proof complexity (see among others: [38, 65, 30, 59, 62, 61, 15, 24, 16, 14, 6, 3, 54]). The $\mathrm{(L)OP}_n$ formulas encode the negation of the (Linear) Ordering Principle which asserts that each finite (linearly) ordered set has a maximal element and was introduced and studied, among others, in the works [45, 67, 23].

Our work spans different proof systems. In fact, they are all actually refutation systems, though we often use the terms interchangeably.

2.1. Resolution and Res(s).

$\mathrm{Res}(s)$ is a refutational proof system extending Resolution to s-bounded DNFs, introduced by Krajíček in [44]. As a generalisation of Resolution, the complexity of proofs in $\mathrm{Res}(s)$ for the unary encoding was largely analysed in several works [6, 31, 33, 65, 1, 60].

A principal motivation for the present work is to approach size lower bounds of refutations in Resolution for families of contradictions in the usual unary encoding, by looking at the complexity of proofs in $\mathrm{Res}(s)$ for the corresponding families of contradictions where witnesses are given in the binary encoding. This method is justified by our observation, specified in Lemmas 4.1 and 5.1, that for a family of contradictions encoding a principle which is expressible as a Π_2 first-order formula having no finite models, short $\mathrm{Res}(\log n)$ refutations of their binary encoding can be obtained from short Resolution refutations for the unary encoding. In light of this observation we begin with the study of the binary version of the k-Clique Formula. Indeed a significant size lower bound for the unary version of the k-Clique Formulas in full Resolution is a long-standing open problem. At present such lower bounds are known only for restrictions of Resolution: in the treelike case [17], and, in a recent major breakthrough, for the case of read-once (or regular) Resolution [5].

2.2. Sherali-Adams.

It is well-known that questions on the satisfiability of propositional CNF formulas may be reduced to questions on feasible solutions for certain Integer Linear Programs (ILPs). In light of this, several ILP-based proof (more accurately, refutation) systems have been suggested for propositional CNF formulas, based on proving that the relevant ILP has no solutions. Typically, this is accomplished by relaxing an ILP to a continuous Linear Program (LP), which itself may have (non-integral) solutions, and then reconstraining this LP iteratively until it has a solution iff the original ILP had a solution (which happens at the point the LP has no solution). Among the most popular ILP-based refutation systems are Cutting Planes [36, 25] and several others proposed by Lovász and Schrijver [53].

Another method for solving ILPs was proposed by Sherali and Adams [66], and was introduced as a propositional refutation system in [26]. Since then it has been considered as a refutation system in the further works [29, 9]. The Sherali-Adams system (SA) is of significant interest as a static variant of the Lovász-Schrijver system without semidefinite cuts (LS). It is proved in [49] that the SA *rank* of a polytope, roughly speaking the number of iterations the polytope is reconstrained until it becomes empty, is less than or equal to its LS rank; hence we may claim that with respect to rank SA is at least as strong as LS (though it is unclear whether it is strictly stronger).

The binary encoding implicitly enforces an at-most-one constraint on the witness at the same time as it does the at-least-one. That is, it specifies a unique witness. Another way to enforce this is with unary functional constraints of the form $v_{\mathbf{a},1} + \ldots + v_{\mathbf{a},n} = 1$ (cf. the unary functional encoding of Section 2.6), where $\mathbf{a}$ comes from a sequence of universal variables preceding the single existential variable the sum is witnessing. This contrasts with the standard unary encoding which would be of the form $v_{\mathbf{a},1} + \ldots + v_{\mathbf{a},n} \geq 1$. We paraphrase our new variant as being (the unary) *encoding with equalities* or "SA-with-equalities" and study this variant explicitly.

2.3. SA+Squares. We continue by considering a refutation system we call SA+Squares which is between SA and Lasserre (Sum-of-Squares) [48] (see also [49] for comparison between these systems). SA+Squares appears as Static-LS$^\infty_+$ in [37], where SA is denoted Static-LS$^\infty$. In this system one can always assume the non-negativity of (the linearisation of) any squared polynomial. In contrast to our system SA-with-equalities, we will see that the rank of the unary encoding of the Pigeonhole Principle is 2, while the rank of the Ordering Principle is linear. We prove this by showing a certain moment matrix is positive semidefinite.

2.4. Three combinatorial principles. We will now delve more deeply into known and new results for our three combinatorial principles. These are depicted in a visually agreeable fashion in Tables 1 and 2. The principles themselves will be introduced in the appropriate section, though there is a table at the end of the appendix in which they can be conveniently found together in both the unary and binary encodings. Let us adopt the following convention, which we will exemplify with the Pigeonhole Principle. PHP refers to the principle (independently of the coding of the witnesses), PHP^m_n refers to the unary encoding and Bin-PHP^m_n refers to the binary encoding.

Res(s)	unary	binary
(Bin-)$\mathrm{Clique}_{\mathrm{n}}^{\mathrm{k}}$	open	not fpt $n^{\Omega(k)/\mathrm{d}(s)}$ Corollary 4.9
(Bin-)PHP_n^m	subexponential upper $2^{O(\sqrt{n \log n})}$ [24]	almost exponential lower $2^{n^{1-\delta}}$ Theorem 5.8
(Bin-)OP_n	polynomial upper $O(n^3)$ [67]	polynomial upper $O(n^3)$ Lemma 9.2

SA size	unary	binary
(Bin-)PHP_n^{n+1}	quadratic upper $O(n^2)$ [63]	exponential tight $2^{\Theta(n)}$ Corollary 6.6

SA rank	unary	binary
(Bin-)LOP_n	linear tight $n-2$ [29]	logarithmic upper $2 \log n$ Corollary 7.3

TABLE 1

Comparison of proof complexity between unary and binary encodings. In the first table, $\mathrm{d}(s)$ *is a doubly exponential function and consider* m *to be exponential in* n*. A fixed parameter tractable (fpt) complexity takes the form of* $f(k)n^{O(1)}$ *and is ruled out by our result for* $\text{Bin-Clique}_{\mathrm{n}}^{\mathrm{k}}$ *in* $\mathrm{Res}(s)$*.*

unary rank	SA	SA-with-equalities	SA+Squares	Lasserre
PHP_n^{n+1}	linear tight [29]	linear tight [29]	constant [37]	constant [37]
LOP_n	linear tight [29]	constant Theorem 7.2	linear tight Theorem 8.2	square root almost tight [57]

binary size	SA	SA+Squares	Lasserre
Bin-PHP_n^{n+1}	exponential lower Theorem 6.5	polynomial upper Theorem 8.1	polynomial upper a fortiori
Bin-LOP_n	polynomial upper Corollary 7.3	polynomial upper a fortiori	polynomial upper a fortiori

TABLE 2

A comparison of rank/degree and size for our principles in Sherali-Adams and its relatives. Here by, e.g., 'linear' we mean in the parameter n *parameterising both families, and not the number of variables.*

2.4.1. The k-Clique Formulas. Deciding whether a graph has a k-clique is an important computational problem considered within computer science and its applications. It can be decided in time $n^{O(k)}$ by a brute force algorithm. It is then of the utmost importance to understand whether given algorithmic primitives are sufficient to design algorithms solving the Clique problem more efficiently than the trivial upper bound. Resolution refutations for the formula $\mathrm{Clique}_k^n(G)$ (respectively any CNF F), can be thought of as the execution trace of an algorithm, whose primitives are defined by the rules of the Resolution system, searching for a k-clique inside G (respectively deciding the satisfiability of F). Hence understanding whether there are $n^{\Omega(k)}$ size lower bounds in Resolution for refuting $\mathrm{Clique}_k^n(G)$ would then answer the above question for algorithms based on Resolution primitives. This question was posed in [17] where they proved that for canonical graphs not containing k-cliques, that is $k-1$-partite complete graphs, $\mathrm{Clique}_k^n(G)$ can be refuted efficiently, that is in size $O(n^2 2^k)$. In looking for classes of graphs making hard the formula $\mathrm{Clique}_k^n(G)$ for Resolution, [17] considered the case when G is a random graph obtained by the Erdős-Rényi distribution on graphs. For graphs G in this family, they proved that $\mathrm{Clique}_k^n(G)$ requires $n^{\Omega(k)}$ size refutations in treelike Resolution, obtaining the desired lower bound but only for refutations restricted to tree form. Whether the lower bound for $\mathrm{Clique}_k^n(G)$ holds for general DAG-like Resolution when G is a Erdős-Rényi random graph is a major open problem which motivates this paper and towards which we contribute. This specific problem acquired even more importance as a consequence of two more recent results. On the one hand very recently Atserias et al. in [4] proved an $n^{\Omega(k)}$ lower bound for $\mathrm{Clique}_k^n(G)$ when G is a Erdős-Rényi random graph for the case of *read-once* Resolution refutations, which is a restriction of DAG-like Resolution, where each variable can be resolved at most once along any path in the refutation. On the other hand in the work [51], Lauria et al. consider the binary encoding of Ramsey-type propositional statements, having as a special case a binary version of $\mathrm{Clique}_k^n(G)$: $\text{Bin-Clique}_k^n(G)$. For this binary k-Clique Formula they obtain optimal $n^{\Omega(k)}$ size lower bounds for unrestricted Resolution.

Our Results. We prove (in Corollary 4.9) an $n^{\Omega(k)/\mathrm{d}(s)}$ lower bound for the size of refutations of Bin-Clique_k^n in $\mathrm{Res}(o((\log\log n)^{1/3}))$, where $\mathrm{d}(s)$ is a doubly exponential function and G is a random graph as defined in [17].

2.4.2. The (weak) Pigeonhole Principle. Lower bounds for $\mathrm{Res}(s)$ have appeared variously in the literature for the (weak) Pigeonhole Principle. Of most interest to us are those for the (moderately weak) Pigeonhole Principle PHP_n^{2n}, for $\mathrm{Res}(\sqrt{\log n/\log\log n})$ in [65], improved to $\mathrm{Res}(\epsilon\log n/\log\log n)$ in [60]. Additionally, Buss and Pitassi, in [24], proved an upper bound of $2^{O(\sqrt{n\log n})}$ for the size of refuting PHP_n^m in $\mathrm{Res}(1)$ when $m \geq 2^{\sqrt{n\log n}}$.

In [11], an optimal lower bound is proven for the binary encoding of a relativised version of the pigeon-hole principle in Res(log). Their technique, however, heavily depends on the relativisation and the specific choice of the parameters: no set of αn pigeons out of n^β in total can be consistently mapped onto n holes for any $\alpha,\beta > 1$. Proving a similar lower bound for the standard, unrelativised, version is a big question that remains wide open.

In [29] Dantchev et al. have proved that the SA rank of (the polytope associated with) PHP_n^{n+1} is $n-2$ (where n is the number of holes). That there is a polynomially-sized refutation in SA of PHP_n^{n+1} is noted in [63]. Grigoriev et al. have noted in [37] that there is a rank 2 and polynomially-sized refutation of PHP_n^{n+1} in Lasserre, and it is straightforward to see that this may be implemented in SA+Squares.

Our Results. We prove that in $\mathrm{Res}(s)$, for all $\epsilon > 0$ and $s \leq \log^{\frac{1}{2}-\epsilon}(n)$, the shortest proofs of Bin-PHP_n^m, require size $2^{n^{1-\delta}}$, for any $\delta > 0$ (Theorem 5.8). This is the first size lower bound known for the Bin-PHP^m_n in Res(s). As a by-product of this lower bound we prove a lower bound of the order $2^{\Omega(\frac{n}{\log n})}$ (Theorem 5.4) for the size of the shortest Resolution refutation of Bin-PHP^m_n. Our lower bound for Res(s) is obtained through a technique that merges together the random restriction method, an inductive argument on the s of Res(s) and the notion of *minimal covering* of a k-DNF of [65].

Since we are not using any (even weak) form of Switching Lemma (as for instance in [65, 1]), we consider how tight is our lower bound in Res(s). We prove that Bin-PHP^m_n (Theorem 5.9) can be refuted in size $2^{O(n)}$ in treelike Res(1). This upper bound contrasts with the unary case, PHP^m_n, which instead requires treelike Res(1) refutations of size $2^{\Omega(n \log n)}$, as proved in [16, 30].

For the Pigeonhole Principle, similarly to the k-Clique Principle, we can prove that short Res($\log n$) refutations for Bin-PHP^m_n can be efficiently obtained from short Res(1) refutations of PHP^m_n (Lemma 5.1). This allows us to prove that our lower bound is almost optimal: from the aforementioned result of Buss and Pitassi [24] we deduce an exponential lower bound is not possible for Bin-PHP^m_n in Res($\log n$).

We prove that the binary encoding Bin-PHP^m_n requires exponential size in SA (Theorem 6.5), contrasting with the mentioned polynomially-sized refutations of the unary PHP^m_n. Finally, we prove that Bin-PHP^m_n has polynomially sized and rank 2 refutations in SA+Squares (Theorem 8.1), in line with the corresponding result for the unary Pigeonhole Principle from [37].

2.4.3. Ordering Principles. The Linear ordering formulas LOP$_n$ claim that a linear ordering of some domain has no minimal element. In the case of finite domains, it is false. They were used in [23, 35] as families of formulas witnessing the optimality of the size-width tradeoffs for Resolution ([15]), so that they require high width to be refuted, but still admit polynomial size refutations in Resolution. If we drop the stipulation that the order is linear (total), we call the principle OP$_n$.

In [29] we showed that the SA rank of (the polytope associated with) LOP$_n$ is $n-2$. Since it is known that SA polynomially simulates Resolution (see e.g. [29]), it follows that there is a polynomially-sized refutation in SA of LOP$_n$. Potechin has proved that LOP$_n$ has refutations in Lasserre of degree $O(\sqrt{n}\log n)$. Though he uses a different version of LOP$_n$ from us, we will see that his upper bound still applies.

Our Results. Firstly, we prove that Bin-OP$_n$ is polynomially provable in Resolution. Secondly, and in the world of SA, we prove that the (unary) encoding of the Ordering Principle with equalities has rank 2 and polynomial size. This allows us to prove that Bin-LOP$_n$ has SA rank at most $2\log n$ and polynomial size. We prove a rank lower bound in SA+Squares for LOP$_n$ of $\Omega(n)$, thus giving a quadratic separation in terms of rank between SA+Squares and Lasserre.

2.5. Main technical contributions. As observed, one of the main contributions of this work is the $n^{\Omega(k)/\mathrm{d}(s)}$ size lower bounds for Res(s) refutations of Bin-Clique$^n_k(G)$ when G is a random graph as, for example, defined in [17]. The interest of this lower bound lies in the fact that the Resolution complexity of Clique$^n_k(G)$ at present is unknown and, as we prove in this paper, this lower bound would follow from a meaningful lower bound for Bin-Clique$^n_k(G)$ in Res($\log n$). Our result for Res(s) for Bin-Clique$^n_k(G)$ hence contributes towards this goal.

The main mathematical tool used so far to prove size lower bounds in Res(s) is a simplified version of the Håstad Switching Lemma [40] which was introduced in the work of Segerlind, Buss and Impagliazzo [65] and later used (and slightly improved in [60]) in all other works proving size lower bounds for Res(s) [1]. Only for Res(2), in the work [6], there is an example of a size lower bound using a random restriction method inherited from Resolution.

In this work we devise a recursive method to prove size lower bounds in Res(s), which is especially suitable for binary principles and runs by recursion from s to 1. Contrary to previous methods, our method does not use any form of the Håstad Switching Lemma. The main ingredients of our approach are: (1) special classes of random restrictions, which are especially suited for binary principles and can be easily composed recursively; (2) the notion of *covering number* for a DNF (that is the minimal number of literals covering all the terms of a DNF), which was introduced in [65]. The high level idea of the lower bound proof is as follows. Setting the covering number in the proper way, the recursion process applied on an allegedly small refutation of a binary principle in Res(s) ends with a small Res(1), that is Resolution, refutation of a simplification of the same principle defined on a smaller but still meaningful domain. At this point it is sufficient to prove (or to use if known) a size lower bound for the principle in Resolution.

The lower bound for the k-Clique Formulas in Res(s) is obtained by capturing a hardness property for the k-Clique Formulas which closely follows those defined in [17] for the unary case and later used and extended in [52, 4]. However, differently from previous lower bounds, we isolate the hardness property in a definition (see Definition 4.2) and a lemma called the *Extension Lemma* (see Lemma 4.3), whose aim is that of capturing the existence of non-trivial families of partial assignments that applied to the k-Clique Formula do not trivialise its Resolution refutations. This is inspired by the Atserias-Dalmau [7] approach to prove width lower bounds (and hence size lower bounds) for Resolution.

2.6. Contrasting unary and binary principles. We go on to consider the relative properties of unary and binary encodings, especially for Resolution. We take the case in which the principle is binary and involves total comparison on all its relations. That is, where there are axioms of the form $v_{i,j} \oplus v_{j,i}$, where $\oplus$ indicates XOR, for each $i \neq j$. We argue that the proof complexity in Resolution of such principles will not increase significantly (by more than a polynomial factor) when shifting from the unary encoding to the binary encoding.

The *unary functional* encoding of a combinatorial principle replaces the big disjunctive clauses of the form $v_{i,1} \vee \ldots \vee v_{i,n}$, with $v_{i,1} + \ldots + v_{i,n} = 1$, where addition is made on the natural numbers. We already met this in the context of SA, but it is equivalent to augmenting the axioms $\neg v_{i,j} \vee \neg v_{i,k}$, for $j \neq k \in [n]$. One might argue that the unary functional encoding is the true unary analog to the binary encoding, since the binary encoding naturally enforces that there is a single witness alone. It is likely that the non-functional formulation was preferred for its simplicity (similarly as the Pigeonhole Principle is often given in its non-functional formulation).

In Subsection 9.1, we prove that the Resolution refutation size increases by only a quadratic factor when moving from the binary encoding to the unary functional encoding. This is interesting because the same does not happen for treelike Resolution, where the unary encoding of the Pigeonhole Principle has complexity $2^{\Theta(n \log n)}$ [16, 30], while, as we prove in Subsection 5.1 (Theorem 5.9), the binary (functional) encoding is $2^{\Theta(n)}$. The unary encoding complexity is noted in [31] and remains true for the unary functional encoding with the same lower-bound proof. The binary encoding complexity is addressed directly in this paper.

2.7. Structure of the paper. After the preliminaries in Section 3, we move on to the Res(s) lower bounds for Bin-Clique^n_k in Section 4 and Bin-PHP^m_n in Section 5.

In Section 6 we prove our SA size lower bound for Bin-PHP^m_n and in Section 7 we prove our SA size and rank upper bounds for the Linear Ordering Principle with equalities, which apply, as a corollary, also to to Bin-LOP$_n$. In Section 8, we introduce SA+Squares and discuss upper bounds for PHP and give a lower bound for LOP$_n$. In Section 9, we make further comments on the contrasts between unary and binary encodings in general for Resolution. In Section 10, we make some final remarks.

Two objects inhabit an appendix. Firstly, an argument that Potechin's Lasserre upper bound for LOP$_n$ from [57] applies also to our encoding. Secondly, a table recapping the unary and binary encodings of the main principles.

3. Preliminaries. Let $[n]$ be the set $\{1, \ldots, n\}$. Let us assume, without loss of much generality, that n is a power of 2. Cases where n is not a power of 2 are handled in the binary encoding by explicitly forbidding possibilities. Let $\mathrm{bin}(a)$ be the sequence $a_1 \ldots a_{\log n}$, which is a written in binary, say from the most significant digit to the least.

If v is a propositional variable, then $v^0 = \neg v$ indicates the negation of v, while v^1 indicates v. We denote by $\top$ and $\bot$ the Boolean values "true" and "false", respectively. A *literal* is either a propositional variable or a negated variable. We will denote literals by small letters, usually l's. An *s-conjunction* (*s-disjunction*) is a conjunction (disjunction) of at most s literals. A *clause* with s literals is an s-disjunction. The width $w(C)$ of a clause C is the number of literals in C. A *term* (*s-term*) is either a conjunction (s-conjunction) or a constant, $\top$ or $\bot$. An *s-DNF* or *s-clause* (*s-CNF*) is a disjunction (conjunction) of an unbounded number of s-conjunctions (s-disjunctions). We will use calligraphic capital letters to denote s-CNFs or s-DNFs, usually $\mathcal{C}$s for CNFs, $\mathcal{D}$s for DNFs and $\mathcal{F}$s for both. For example, $((v_1 \wedge \neg v_2) \vee (v_2 \wedge v_3) \vee (\neg v_1 \wedge v_3))$ is an example of a 2-DNF and its negation $((\neg v_1 \vee v_2) \wedge (\neg v_2 \vee \neg v_3) \wedge (v_1 \vee \neg v_3))$ is an example of a 2-CNF.

3.1. Res(s) and Resolution. We can now describe the propositional refutation system $\mathrm{Res}\,(s)$ ([43]). It is used to *refute* (i.e. to prove inconsistency) of a given set of s-clauses by deriving the empty clause from the initial clauses. There are four derivation rules:

1. The $\wedge$-*introduction rule* is

$$\frac{\mathcal{D}_1 \vee \bigwedge_{j \in J_1} l_j \quad \mathcal{D}_2 \vee \bigwedge_{j \in J_2} l_j}{\mathcal{D}_1 \vee \mathcal{D}_2 \vee \bigwedge_{j \in J_1 \cup J_2} l_j},$$

provided that $|J_1 \cup J_2| \leq s$.

2. The *cut (or resolution) rule* is

$$\frac{\mathcal{D}_1 \vee \bigvee_{j \in J} l_j \quad \mathcal{D}_2 \vee \bigwedge_{j \in J} \neg l_j}{\mathcal{D}_1 \vee \mathcal{D}_2}.$$

3. The two *weakening rules* are

$$\frac{\mathcal{D}}{\mathcal{D} \vee \bigwedge_{j \in J} l_j} \quad \text{and} \quad \frac{\mathcal{D} \vee \bigwedge_{j \in J_1 \cup J_2} l_j}{\mathcal{D} \vee \bigwedge_{j \in J_1} l_j},$$

provided that $|J| \leq s$.

A $\mathrm{Res}(s)$ refutation can be considered as a directed acyclic graph (DAG), whose sources are the initial clauses, called also axioms, and whose only sink is the empty clause. We shall define *the size of a proof* to be the number of internal nodes of the graph, i.e. the number of applications of a derivation rule, thus ignoring the size of the individual s-clauses in the refutation. In principle the s from "Res(s)" could depend on n — an important special case is $\mathrm{Res}(\log n)$.

Clearly, Res(1) is *(ordinary) Resolution*, working on clauses, and using only the cut rule, which becomes the usual resolution rule, and the first weakening rule. Given an unsatisfiable CNF $\mathcal{C}$, and a Res(1) refutation π of $\mathcal{C}$ the width of π, $w(\pi)$, is the maximal width of a clause in π. The width of refuting $\mathcal{C}$ in Res(1), $w(\vdash \mathcal{C})$, is the minimal width over all Res(1) refutations of $\mathcal{C}$.

A *covering set* for an s-DNF $\mathcal{D}$ is a set of literals L such that each term of $\mathcal{D}$ has at least one literal in L. The *covering number* $c(\mathcal{D})$ of an s-DNF $\mathcal{D}$ is the minimal size of a covering set for $\mathcal{D}$. We extend the definition of covering number to the case of s-CNFs: the covering number of a s-CNF F is the covering number of the DNF obtained by applying De Morgan simplifications to $\neg F$.

Let $\mathcal{F}(v_1 \ldots, v_n)$ be a boolean s-DNF (resp. s-CNF) defined over variables $V = \{v_1, \ldots, v_n\}$. A *partial assignment* ρ to $\mathcal{F}$ is a truth-value assignment to some of the variables of $\mathcal{F}$: $dom(\rho) \subseteq V$. By $\mathcal{F}\restriction_\rho$ we denote the formula $\mathcal{F}'$ over variables in $V \setminus dom(\rho)$ obtained from $\mathcal{F}$ after simplifying in it the variables in $dom(\rho)$ according to the usual boolean simplification rules of clauses and terms.

Similarly to what was done for treelike Res(s) refutations in [33], if we turn a Res(s) refutation of a given set of s-clauses $\mathcal{F}$ upside-down, i.e. reverse the edges of the underlying graph and negate the s-clauses on the vertices, we get a special kind of restricted branching s-program whose nodes are labelled by s-CNFs and at each node some s-disjunction is questioned. The restrictions placed on the branching program are as follows.

Each vertex is labelled by an s-CNF which partially represents the information that can be obtained along any path from the source to the vertex (this is a *record* in the parlance of [58]). Obviously, the (only) source is labelled with the constant $\top$. There are two kinds of queries that can be made by a vertex:

1. Querying a new s-disjunction, and branching on the answer, which can be depicted as follows.

$$\begin{tikzcd}[column sep=small] & \begin{array}{c}\mathcal{C}\\ ?\bigvee_{j\in J} l_j\end{array} \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\ \mathcal{C}\wedge\bigvee_{j\in J} l_j & & \mathcal{C}\wedge\bigwedge_{j\in J}\neg l_j \end{tikzcd} \tag{3.1}$$

2. Querying a known s-disjunction, and splitting it according to the answer:

$$\begin{tikzcd}[column sep=small] & \begin{array}{c}\mathcal{C}\wedge\bigvee_{j\in J_1\cup J_2} l_j\\ ?\bigvee_{j\in J_1} l_j\end{array} \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\ \mathcal{C}\wedge\bigvee_{j\in J_1} l_j & & \mathcal{C}\wedge\bigvee_{j\in J_2} l_j \end{tikzcd} \tag{3.2}$$

There are two ways of forgetting information,

$$\begin{array}{c}\mathcal{C}_1\wedge\mathcal{C}_2\\ \downarrow\\ \mathcal{C}_1\end{array} \quad\text{and}\quad \begin{array}{c}\mathcal{C}\wedge\bigvee_{j\in J_1} l_j\\ \downarrow\\ \mathcal{C}\wedge\bigvee_{j\in J_1\cup J_2} l_j\end{array}, \tag{3.3}$$

the point being that forgetting allows us to equate the information obtained along two different branches and thus to merge them into a single new vertex. For simplicity when calculating the size of refutation subtrees, let us assume that a weakening may be integrated into either side of a query. A sink of the branching s-program must be labelled with the negation of an s-clause from $\mathcal{F}$. Thus the branching s-program is supposed by default to solve the *Search Problem for* $\mathcal{F}$: given an assignment of the variables, find a clause which is falsified under this assignment.

The equivalence between a $\mathrm{Res}(s)$ refutation of $\mathcal{F}$ and a branching s-program of the kind above is obvious. Naturally, if we allow querying single variables only, we get branching 1-programs – decision DAGs – that correspond to Resolution. If we do not allow the forgetting of information, we will not be able to merge distinct branches, so what we get is a class of decision trees that correspond precisely to the treelike version of these refutation systems. The queries of the form (3.1) and (3.2) as well as forget-rules of the form (3.3) give rise to a Prover-Adversary game (see [58] where this game was introduced for Resolution). In short, Adversary claims that $\mathcal{F}$ is satisfiable, and Prover tries to expose him. Prover always wins if her strategy is kept as a branching program of the form we have just explained, whilst a good (randomised) Adversary's strategy would show a lower bound on the branching program, and thus on any $\mathrm{Res}\,(s)$ refutation of $\mathcal{F}$.

LEMMA 3.1. *If a CNF ϕ has a refutation in $\mathrm{Res}(k+1)$ of size N, whose corresponding branching $(k+1)$-program has no $(k+1)$-CNFs of covering number $\geq d$, then ϕ has a $\mathrm{Res}(k)$ refutation of size $2^{d+1} \cdot N$ (which is $\leq e^d \cdot N$ when $d > 4$).*

Proof. In the branching program, consider a $(k+1)$-CNF ϕ whose covering number $< d$ is witnessed by variable set $V' := \{v_1, \ldots, v_{d-1}\}$. At this node some $(k+1)$-disjunction $(l_1 \vee \ldots \vee l_k \vee l_{k+1})$ is questioned.

Now in place of the CNF record ϕ in our original branching program we expand a mini-tree of size 2^{d+1} with 2^d leaves questioning all the variables of V' as well as the literal l_{k+1}. Clearly, each evaluation of these reduces ϕ to a k-CNF that logically implies ϕ. This may involve a weakening step in the corresponding $\mathrm{Res}(k)$ refutation. It remains to explain how to link the leaves of these mini-trees to the roots of other mini-trees. At each leaf we look to see whether we have the information l_{k+1} or $\neg l_{k+1}$. If l_{k+1} then we link immediately to the root of the mini-tree corresponding to the yes-answer to $(l_1 \vee \ldots \vee l_k \vee l_{k+1})$ (without asking a question). If $\neg l_{k+1}$ then we question $(l_1 \vee \ldots \vee l_k)$ and, if this is answered yes, link the yes-answer to $(l_1 \vee \ldots \vee l_k \vee l_{k+1})$, otherwise to its no-answer. ∎

3.2. Sherali-Adams via (integer) linear programming. Following [29] we define the SA proof system in a ILP form and hence in terms of linear inequalities and we explain later the equivalence with an alternative definition by polynomials.

Let $\mathcal{C}$ be a CNF $C_1 \wedge \ldots \wedge C_m$ in variables $V = \{v_1, \ldots, v_n\}$. Let $L_V = \{v_1, \ldots, v_n, \neg v_1, \ldots, \neg v_n\}$ and adopt the convention that for $l \in L_V$, if $l = \neg v$ then $\overline{l} = v$ and if $l = v$, then $\overline{l} = \neg v$. First we introduce a set of integer variables of the form Z_D, where D is a conjunction of *distinct* literals in L_V, with the meaning that $Z_{\bigwedge_i l_i}$ is false if its subscript is false.*

We consider $Z_1 = Z_\emptyset$, where $\emptyset$ is an empty conjunction, to be associated with the monomial equation $0 = 0$ and we assume that the *names* of the Z variables fulfil the basic properties of the $\wedge$ operator such as commutativity and idempotence. So, for

*We are considering here n new formal variables $\bar{V} = \{\bar{v}_1, \ldots, \bar{v}_n\}$ such that $v = (1 - \bar{v})$. This allow us to compactly write a polynomial of the form $\prod_i (1 - v_i)$ as a monomial $\prod_i \bar{v}_i$, modulo the set of polynomials stating that $v + \bar{v} = 1$ taken for all variables v.

instance, $Z_{D_1 \wedge D_2}$ is the same variable as $Z_{D_2 \wedge D_1}$, or $Z_{1 \wedge D}$ as well as $Z_{D \wedge D}$ are both the variable Z_D.

For $0 \leq r < 2n$ let $\mathcal{D}_r$ be the set of the conjunctions of at most r literals in L_V (being 1 the empty conjunction). We let $\mathcal{P}_r^{\mathcal{C}}$ to be the polytope specified by the following inequalities.

$$0 \leq Z_{l \wedge D} \leq Z_D \qquad l \in L_V, D \in \mathcal{D}_r \tag{3.4}$$

$$Z_{l \wedge D} + Z_{\bar{l} \wedge D} = Z_D \qquad l \in L_V, D \in \mathcal{D}_r \tag{3.5}$$

$$(Z_{D \wedge l_1} + \cdots + Z_{D \wedge l_k}) \geq Z_D \qquad (l_1 \vee \ldots \vee l_k) \in C, D \in \mathcal{D}_r \tag{3.6}$$

Observe that $\mathcal{P}_0^{\mathcal{C}}$, the polytope associated to $\mathcal{C}$, is specified by the inequalities

$$\begin{cases} 0 \leq Z_l \leq 1 & l \in L_V \\ Z_l + Z_{\bar{l}} = 1 & l \in L_V \\ Z_{l_1} + \cdots + Z_{l_k} \geq 1 & (l_1 \vee \ldots \vee l_k) \in C \end{cases}$$

It is clear that $\mathcal{P}_0^{\mathcal{C}}$ contains integral $\{0, 1\}$ points if and only if $\mathcal{C}$ is satisfiable.

Sherali-Adams (SA) is a static refutation method that takes the polytope $\mathcal{P}_0^{\mathcal{C}}$ whose dimension is $2n$ and *r-lifts* it, by the definition of new variables and constraints, to another polytope $\mathcal{P}_r^{\mathcal{C}}$ whose dimension is $\sum_{\lambda=0}^{r+1} \binom{2n}{\lambda}$. Observe that on unsatisfiable CNFs $\mathcal{C}$, $\mathcal{P}_0^{\mathcal{C}}$ does not contain integral points but it is not necessarily empty, while necessarily $\mathcal{P}_{2n}^{\mathcal{C}}$ is the empty polytope (indeed, already $\mathcal{P}_{n-1}^{\mathcal{C}}$ is empty). Hence the following definition is meaningful.

Definition 3.2. *The* SA-rank *of an unsatisfiable CNF $\mathcal{C}$ (we equivalently say the* SA-rank *of $\mathcal{P}_0^{\mathcal{C}}$) is the minimal $r \leq 2n$ such that $\mathcal{P}_r^{\mathcal{C}}$ is the empty polytope. A* SA-refutation *of $\mathcal{C}$ is a subset of constraints in the definition of $\mathcal{P}_r^{\mathcal{C}}$ that defines an empty polytope.*

Note that SA is polynomially verifiable due to the tractability of linear programming.

Let us point out some simple properties we use later. It is easy to see that for $r' \leq r$, the defining inequalities of $\mathcal{P}_{r'}^{\mathcal{C}}$ are included in those of $\mathcal{P}_r^{\mathcal{C}}$. Hence any solution to the inequalities of $\mathcal{P}_r^{\mathcal{C}}$ gives rise to solutions of the inequalities of $\mathcal{P}_{r'}^{\mathcal{C}}$, when projected onto its variables. If D' is a conjunction of r' literals, then $Z_{D \wedge D'} \leq Z_D$ follows by transitivity from r' instances of (3.4). We refer to the property $Z_{D \wedge D'} \leq Z_D$ as *monotonicity*. Finally, let us note that $Z_{v \wedge \neg v} = 0$ holds in $\mathcal{P}_1^{\mathcal{C}}$ and follows from a single lift of an equality of negation.

Our use of distinct literals Z_v and $Z_{\neg v}$, with the axioms (3.2), is not followed in all expositions of Sherali-Adams as a refutation system SA. Indeed, in [8], the use of these so-called *twin variables* begets a new refutation system labelled SAR (in an apparent homage to the PCR of [2]). Note that the rank measure is equivalent in both versions of SA, and size lower bounds, for our version with twin variables, are at least as strong as with the alternative version.

3.2.1. Sherali-Adams via polynomials. Here we give an alternative definition of Sherali-Adams and explain its relation to the one just given.

Definition 3.3. *A* Sherali-Adams *refutation of a set of linear inequalities $a_1 \geq 0, \ldots, a_m \geq 0$ over a set of variables V is a formal equality of the form*

$$c_0 + \sum_{i=1}^{m} c_i a_i = -1 \tag{3.7}$$

where each c_i is a polynomial over V with non-negative coefficients, and the multiplication is carried out over the quotient ring $\mathbb{R}^V/\{v^2 - v : v \in V\}$ (that is, idempotently). The degree *of the refutation is the maximum degree of the polynomials $c_i a_i$. The* size *of the refutation is the total number of monomials appearing with nonzero coefficients on the left hand side of* (3.7)

It is clear that Sherali-Adams is sound, in the sense that if a set of linear inequalities admits a Sherali-Adams refutation then it has no 0/1 solutions. Once the degree is fixed, the search for the coefficients of the c_i in Equation (3.7) can be formulated as a linear program. It can be seen that the dual of this program is exactly the definition given first (see, e.g., [49]). Imagine, for some CNF C over the variables $V = \{v_1, \ldots, v_n\}$ and some rank r, that $\mathcal{P}_r^{\mathcal{C}}$ is nonempty. Then pick some $x \in \mathcal{P}_r^{\mathcal{C}}$ and define a linear operator λ on monomials of degree at most $r+1$ defined by $\lambda(v_{x_1} \cdot v_{x_2} \cdots v_{x_d}) = x(Z_{v_{x_1} \wedge \ldots \wedge v_{x_d}})$. Then the set of inequalities gotten from sending each clause $l_1 \vee \ldots \vee l_k$ in $\mathcal{C}$ to $\sum_{i=1}^k l_i \geq 1$ has no Sherali-Adams refutation of degree at most r, because then λ when applied to both sides of (3.7) would produce a contradiction.

4. Res(s) and the binary encoding of k-Clique. Consider a graph G such that G is formed from k blocks of n nodes each: $G = (\bigcup_{b \in [k]} V_b, E)$, where edges may only appear between distinct blocks. Thus, G is a k-partite graph. Let the edges in E be denoted as pairs of the form $E((i,a),(j,b))$, where $i \neq j \in [k]$ and $a, b \in [n]$.

The (unary) k-Clique CNF formulas $\text{Clique}_k^n(G)$ has variables $v_{i,q}$ with $i \in [k], q \in [n]$, with clauses $\neg v_{i,a} \vee \neg v_{j,b}$ whenever $\neg E((i,a),(j,b))$ (i.e. there is no edge between node a in block i and node b in block j), and clauses $\bigvee_{a \in [n]} v_{i,a}$, for each block i. This expresses that G has a k-clique (with one vertex in each block), which we take to be a contradiction, since we will arrange for G not to have a k-clique. Notice that this formula encodes the fact that the graph contains a *transversal k-clique*, that is, a k-clique in which each node belongs to a different block. As noticed in [17, 4] a graph can contain a k-clique but no transversal k-clique for a given partition. Finding a transversal k-clique in a given graph is intuitively more difficult then finding a k-clique, hence proving that a graph does not contain a transversal k-clique should be easier than proving it does not contain any k-clique. This was formally proved to hold even for treelike Resolution (see Lemma 2.2 in [4]).

$\text{Bin-Clique}_k^n(G)$ variables $\omega_{i,j}$ range over $i \in [k], j \in [\log n]$. Let us assume for simplicity of our exposition that n is a power of 2, the general case requires the explicit forbidding of certain combinations. Let $a \in [n]$ and let $a_1 \ldots a_{\log n}$ be $\text{bin}(a)$. Each (unary) variable $v_{i,a}$ semantically corresponds to the conjunction $(\omega_{i,1}^{a_1} \wedge \ldots \wedge \omega_{i,\log n}^{a_{\log n}})$, where

$$\omega_{i,j}^{a_j} = \begin{cases} \omega_{i,j} & \text{if } a_j = 1 \\ \neg\omega_{i,j} & \text{if } a_j = 0 \end{cases}$$

Hence in $\text{Bin-Clique}_k^n(G)$ we encode the unary clauses $\neg v_{i,a} \vee \neg v_{j,b}$, by the clauses

$$(\omega_{i,1}^{1-a_1} \vee \ldots \vee \omega_{i,\log n}^{1-a_{\log n}}) \vee (\omega_{j,1}^{1-b_1} \vee \ldots \vee \omega_{j,\log n}^{1-b_{\log n}}).$$

Notice that the wide clauses $\bigvee_{a \in [n]} v_{i,a}$ from the unary encoding automatically become true under the binary encoding.

By the next lemma short Resolution refutations for $\text{Clique}_k^n(G)$ can be translated into short $\text{Res}(\log n)$ refutations of $\text{Bin-Clique}_k^n(G)$. Hence to obtain lower bounds for $\text{Clique}_k^n(G)$ in Resolution, it suffices to obtain lower bounds for $\text{Bin-Clique}_k^n(G)$ in $\text{Res}(\log n)$.

Lemma 4.1. *Suppose there are Resolution refutations of* $\text{Clique}_k^n(G)$ *of size* S. *Then there are* $\text{Res}(\log n)$ *refutations of* $\text{Bin-Clique}_k^n(G)$ *of size* S.

Proof. Where the decision DAG for $\text{Clique}_k^n(G)$ questions some variable $v_{i,a}$, the decision branching $\log n$-program questions instead $(\omega_{1,1}^{1-a_1} \vee \ldots \vee \omega_{1,\log n}^{1-a_{\log n}})$ where the out-edge marked true in the former becomes false in the latter, and vice versa. What results is indeed a decision branching $\log n$-program for $\text{Bin-Clique}_k^n(G)$, and the result follows. ∎

Following [17, 4, 51] we consider $\text{Bin-Clique}_k^n(G)$ formulas where G is a random graph distributed according to a variation of the Erdős-Rényi distribution as defined in [17]. In the standard model, random graphs on n vertices are constructed by including every edge independently with probability p. It is known (see for example [19, 20]) that k-cliques appear at the threshold probability p^* approximately equal to $n^{-\frac{2}{k-1}}$: If $p < p^*$, then with high probability there is no k-clique. Following [17, 4, 51] we consider random graphs G on kn vertices where an edge is present between two vertices in distinct blocks with probability $p = n^{-(1+\epsilon)\frac{2}{k-1}}$, for ϵ a constant. We call this distribution $\mathcal{G}_{k,\epsilon}^n(p)$ and we use the notation $G \sim \mathcal{G}_{k,\epsilon}^n(p)$ to say that G is a graph drawn at random from $\mathcal{G}_{k,\epsilon}^n(p)$. In the next sections we explore lower bounds for $\text{Bin-Clique}_k^n(G)$ in $\text{Res}(s)$ for $s \geq 1$, when $G \sim \mathcal{G}_{k,\epsilon}^n(p)$.

4.1. Isolating the properties of G. Let α be a constant such that $0 < \alpha < 1$. Define a set of vertices U in G, $U \subseteq V$ to be an *α-transversal* if: (1) $|U| = \alpha k$, and (2) for all $b \in [k]$, $|V_b \cap U| \leq 1$. Let $B(U) \subseteq [k]$ be the set of blocks mentioned in U, and let $\overline{B(U)} = [k] \setminus B(U)$. We say that U is *extendable* in a block $b \in \overline{B(U)}$ if there exists a vertex $a \in V_b$ that is a common neighbour of all nodes in U, i.e. $a \in N_c(U)$ where $N_c(U)$ is the set of *common neighbours* of vertices in U: $N_c(U) = \{v \in V \mid v \in \bigcap_{u \in U} N(u)\}$.

Let σ be a partial assignment (a restriction) to the variables of $\text{Bin-Clique}_k^n(G)$ and β a constant such that $0 < \beta < 1$. We say σ is *β-total* if σ assigns precisely $\lfloor \beta \log n \rfloor$ bits in each block $b \in [k]$, i.e. $\lfloor \beta \log n \rfloor$ variables $\omega_{b,i}$ in each block b. Note that in general we do not choose the same $\lfloor \beta \log n \rfloor$ bits in each block. Let $v = (i, a)$ be the a-th node in the i-th block in G. We say that a restriction σ is *consistent* with v if for all $j \in [\log n]$, $\sigma(\omega_{i,j})$ is either a_j or not assigned.

Definition 4.2. *Let* $0 < \alpha, \beta < 1$. *An α-transversal set of vertices U is β-extendable, if for all β-total restrictions σ, there is a node v^b in each block $b \in \overline{B(U)}$, such that σ is consistent with v^b.*

An α-transversal is just a set of vertices U comprised of a single vertex from each of αk blocks. It is β-extendable if, for any restriction assigning $\lfloor \beta \log n \rfloor$ bits in each block, there is a vertex adjacent to U in each block outside of U.

Lemma 4.3 (Extension Lemma). *Let* $0 < \epsilon < 1$, *let* $k \leq \log n$. *Let* $1 > \alpha > 0$ *and* $1 > \beta > 0$ *such that* $1 - \beta > 4\alpha(1 + \epsilon)$. *Let* $G \sim \mathcal{G}_{k,\epsilon}^n(p)$. *Over choices of the graph* $\mathcal{G}$, *with probability strictly greater than zero, both the following properties hold:*

1. *all α-transversal sets U are β-extendable;*
2. *$\mathcal{G}$ does not have a k-clique.*

Proof. Let U be an α-transversal set and σ be a β-total restriction. The probability that a vertex w is in $N_c(U)$ is $p^{\alpha k}$. Hence $w \notin N_c(U)$ with probability $(1 - p^{\alpha k})$. After σ is applied, in each block $b \in \overline{B(U)}$ there remain $2^{\log n - \beta \log n} = n^{1-\beta}$ available consistent vertices. Hence the probability that we cannot extend U in each block of

$\overline{B(U)}$ after σ is applied is $(1-p^{\alpha k})^{n^{1-\beta}}$. Fix $c = 2+2\epsilon$ and $\delta = 1-\beta-2\alpha c$. Notice that $\delta > 0$ by our choice of α and β. Since $p = \frac{1}{n^{\frac{c}{k-1}}}$, the previous probability is $(1-1/n^{\alpha c(k/k-1)})^{n^{1-\beta}}$, which is at most $(1-1/n^{2\alpha c})^{n^{1-\beta}}$, which in turn is at most $e^{-\frac{n^{1-\beta}}{n^{2\alpha c}}} = e^{-n^{\delta}}$ (since $e^{-x} = \lim_{m\to\infty}(1-x/m)^m$ and indeed $e^{-x} \geq (1-x/m)^m$ when $x, m \geq 1$).

There are $\binom{k}{\alpha k}$ possible α-transversal sets U and $(\binom{\log n}{\beta \log n} \cdot 2^{\beta \log n})^k$ possible β-total restrictions σ. Let us count the combinations of these:

$$\begin{aligned}\binom{k}{\alpha k} \cdot (\binom{\log n}{\beta \log n} \cdot 2^{\beta \log n})^k &\leq k^{\alpha k} \cdot (\log n)^{\beta k \log n} \cdot 2^{\beta k \log n} \\ &= 2^{\alpha k \log k + \beta k \log n \log\log n + \beta k \log n} \\ &\leq 2^{\log^3 n}.\end{aligned}$$

Note that the last inequality uses $k \leq \log n$. Hence the probability that there is in G an α-transversal set U which is not β-extendable is at most $e^{-n^{\delta}} \cdot 2^{\log^3 n}$ which is tending to zero as n tends to infinity.

To bound the probability that $\mathcal{G}$ contains a k-clique, notice that this probability is bounded above by the expected number of cliques. Now, the expected number of k-cliques can be calculated from the potential maximal number of k-cliques multiplied by the probability that each of these forms a k-clique, that is $n^k p^{\binom{k}{2}} = n^k p^{(k(k-1)/2)}$. Recalling $p = 1/n^{c/k-1}$, we get that the expected number of k-cliques is $n^k n^{-ck/2} = n^{k-ck/2}$. Since $c = 2+2\epsilon$, $k - ck/2 = -\epsilon k$. Hence $n^k n^{-ck/2} = n^{-\epsilon k} \leq n^{-\epsilon}$, which is tending to zero as n tends to infinity.

So the probability that either property (1) or (2) does not hold is bounded above by $2^{\log^3 n} \cdot e^{-n^{\delta}} + n^{-\epsilon}$ which is strictly less than one for sufficiently large n. ∎

4.2. Res(s) **lower bounds for** Bin-Clique$_k^n$**.** Let $s \geq 1$ be an integer. Call a $\frac{1}{2^{s+1}}$-total assignment to the variables of Bin-Clique$_k^n(G)$ an *s-restriction.* A *random s-restriction* for Bin-Clique$_k^n(G)$ is an s-restriction obtained by choosing independently in each block i, $\lfloor \frac{1}{2^{s+1}} \log n \rfloor$ variables among $\omega_{i,1}, \ldots, \omega_{i,\log n}$, and setting these uniformly at random to 0 or 1.

Let $s, k \in \mathbb{N}$, $s, k \geq 1$ and let $G \sim \mathcal{G}^n_{k,\epsilon}(p)$ be a graph over nk nodes and k blocks which does not contain a k-clique. Fix $\delta = \frac{1}{2 \cdot 96^2}$ and $\mathrm{p}(s) = 2^{s^2+3s}$ and $\mathrm{d}(s) = (\mathrm{p}(s)s)^s$.

Let Bin-Clique$_k^n(G)\restriction_\rho$ denote Bin-Clique$_k^n(G)$ restricted by ρ. Consider the following property.

Definition 4.4. *We say that* property Clique(G, s, k) *holds if for any s-restriction ρ, there are no Res(s) refutations of* Bin-Clique$_k^n(G)\restriction_\rho$ *of size less than* $n^{\frac{\delta(k-1)}{\mathrm{d}(s)}}$.

If the property Clique(G, s, k) holds, we immediately have an $n^{\Omega(k)}$ size lower bound for refuting Bin-Clique$_k^n(G)$ in Res(s) (if we view s as a constant).

Corollary 4.5. *Let s, k be integers, $s \geq 1, k > 1$. Let G be a graph and assume that* Clique(G, s, k) *holds. Then there are no Res(s) refutations of* Bin-Clique$_k^n(G)$ *of size smaller than* $n^{\delta\frac{k-1}{\mathrm{d}(s)}}$.

Proof. Choose ρ to be any s-restriction. The result follows from the previous definition since the shortest refutation of a restricted principle can never be larger than the shortest refutation of the unrestricted principle. ∎

We use the previous corollary to prove lower bounds for $\text{Bin-Clique}_k^n(G)$ in $\text{Res}(s)$ as long as $s \in o((\log\log n)^{\frac{1}{3}})$.

THEOREM 4.6. *Let k be an integer with $k > 1$, and $s > 1$ be an integer with $s \in o((\log\log n)^{\frac{1}{3}})$. Then there exists a graph G such that all Res(s) refutations of* $\text{Bin-Clique}_k^n(G)$ *have size at least $n^{\Omega(k)/\mathrm{d}(s)}$.*

Proof. Let $\beta = \frac{3}{4}$ and $\alpha = \frac{1}{16(1+2\epsilon)}$. Let $0 < \epsilon < 1$ be given. It follows that $1 - \beta > 4\alpha(1+\epsilon)$ holds.

By Lemma 4.3, we can fix $G \sim \mathcal{G}_{k,\epsilon}^n$ such that:

1. all α-transversal sets U are β-extendable;
2. $\mathcal{G}$ does not have a k-clique.

We will prove, by induction on s (while $s \in o((\log\log n)^{\frac{1}{3}})$, that property $\text{Clique}(G, s, k)$ does hold. Lemma 4.7 is the base case and Lemma 4.8 the inductive case. The result then follows by Corollary 4.5. ∎

LEMMA 4.7 (Base Case). $\text{Clique}(G, 1, k)$ *does hold.*

Proof. Fix $\beta = \frac{3}{4}$ and $\alpha = \frac{1}{16(1+2\epsilon)}$. Note that $\frac{1}{16} > \alpha > \frac{1}{48}$ and $\mathrm{d}(1) = 16$. Notice also that $1 - \beta > 4\alpha(1+\epsilon)$ holds.

Let ρ be a 1-restriction, that is, a $\frac{1}{4}$-total assignment. We claim that any Resolution refutation of $\text{Bin-Clique}_k^n(G)\restriction_\rho$ must have width at least $\frac{k\log n}{96}$. This is a consequence of Property 1 of the Extension Lemma (4.3), which we henceforth abbreviate as the *extension property*, which allows Adversary to play against Prover with the following strategy. For each block, while fewer than $\frac{\log n}{2}$ bits are known, Adversary offers Prover a free choice. Once $\frac{\log n}{2}$ bits are set, then Adversary chooses an assignment for the remaining bits according to the extension property. Summing up the $\frac{1}{4}$ (proportion of bits in the $\frac{1}{4}$-total assignment) with a potential further $\frac{1}{2}$ of the bits set in the game, we obtain no more than $\frac{3}{4} = \beta$ proportion of bits set, in each block (though the bits set in each block need not be the same). Using the extension property separately in each block, we can guarantee that an appropriate assignment to the remaining bits also exists. Since we can do this over $\alpha k > \frac{k}{48}$ blocks, this allows the game to continue until some CNF record has width at least $\frac{\log n}{2} \cdot \frac{k}{48} = \frac{k\log n}{96}$. Size-width tradeoffs for Resolution [15] tell us that minimal size to refute any unsatisfiable CNF F is lower bounded by $2^{\frac{(w(\vdash F)-w(F))^2}{16V(F)}}$†. In our case $w(F) = 2\log n$ and $V(F) = k\log n$, hence the minimal size required is $\geq 2^{\frac{(\frac{k\log n}{96}-2\log n)^2}{16k\log n}} = 2^{\frac{\log n(\frac{k}{96}-2)^2}{16k}} = n^{\frac{(\frac{k}{96}-2)^2}{16k}}$. It is not difficult to see that $\frac{(\frac{k}{96}-2)^2}{16k} > \frac{(k-1)}{2\cdot 16\cdot 96^2}$ when $k > 2\cdot 16\cdot 96^2$. Since $\delta = \frac{1}{2\cdot 96^2}$ and $\mathrm{d}(1) = 16$ the result is proved.

For short, let $L(s) := n^{\frac{\delta(k-1)}{\mathrm{d}(s)}}$ denote the size bound from Definition 4.4.

LEMMA 4.8 (Inductive Case). *Let $s \in o((\log\log n)^{\frac{1}{3}})$. Then* $\text{Clique}(G, s-1, k)$ *implies* $\text{Clique}(G, s, k)$.

Proof. Assume (towards a contradiction) the opposite – that $\text{Clique}(G, s-1, k)$ holds but there is some s-restriction ρ such that $\text{Bin-Clique}_k^n(G)\restriction_\rho$ has a refutation π of size strictly less than $L(s)$. Fix c to be such that

$$2^{c+2} = \frac{L(s-1)}{L(s)}.$$

†According to [46] Th 8.11

Define $r = \frac{c}{s}$ and let us call a *bottleneck* a CNF record R in π whose covering number is $\geq c$. Hence in such a CNF record it is always possible to find r pairwise disjoint s-tuples of literals $T_1 = (\ell_1^1, \ldots, \ell_1^s), \ldots, T_r = (\ell_r^1, \ldots, \ell_r^s)$ such that the $\bigwedge T_i$'s are among the terms of the s-DNF forming the CNF record R.

Let σ be a *random s-restriction* on the variables of $\text{Bin-Clique}_k^n(G){\restriction}_\rho$. Let us say that σ *kills a tuple* T if it sets to 0 all literals in T (remember that a record s-CNF is the negation of a s-DNF) and that T *survives* σ otherwise, and let us say that σ *kills* R if it kills at least one of the tuples in R. Let Σ_i be the event that T_i survives σ and Σ_R the event that R survives σ. We claim (postponing the proof) that

Claim 1. *If R is a bottleneck, then* $\Pr[\Sigma_R] \leq (1 - \frac{1}{\mathrm{p}(s)})^r$.

Consider now the restriction $\tau = \rho\sigma$. This is an $(s-1)$-restriction on the variables of $\text{Bin-Clique}_k^n(G)$. We argue that in $\pi{\restriction}_\tau$, with probability more than zero, there is no bottleneck. Notice that by the union bound the probability that there exists such a bottleneck CNF record R that survives in $\pi{\restriction}_\tau$, is bounded by

$$\Pr[\exists R \in \pi{\restriction}_\rho \colon \Sigma_R] \leq |\pi{\restriction}_\tau| \left(1 - \frac{1}{\mathrm{p}(s)}\right)^r.$$

(Recall that the probabilistic aspect here comes from σ being a random s-restriction.) We claim that this probability is < 1. Notice that $(1 - \frac{1}{\mathrm{p}(s)})^r \leq e^{-\frac{c}{s\,\mathrm{p}(s)}}$ using the definition of r. So to prove the claim it is sufficient to prove that $|\pi{\restriction}_\tau| < e^{\frac{c}{\mathrm{p}(s)s}}$. As $|\pi{\restriction}_\tau| \leq |\pi{\restriction}_\rho|$ and as by assumption $|\pi{\restriction}_\rho| \leq L(s)$ we can show instead that

$$e^{\frac{c}{s\cdot\mathrm{p}(s)}} > L(s)$$

or equivalently that $e^c \geq L(s)^{s\cdot\mathrm{p}(s)}$. Now, as c is increasing (in n - see the discussion following the conclusion of this proof) we have, for n large enough,

$$e^c > 2^{c+2} = \frac{L(s-1)}{L(s)}$$

so what we will show instead is that

$$L(s-1) \geq L(s)^{s\cdot\mathrm{p}(s)+1} \tag{4.1}$$

$$\Leftrightarrow n^{\frac{\delta(k-1)}{((s-1)\cdot\mathrm{p}(s-1)))^{s-1}}} \geq \left(n^{\frac{\delta(k-1)}{(s\cdot\mathrm{p}(s))^s}}\right)^{s\cdot\mathrm{p}(s)+1} \tag{4.2}$$

$$\Leftrightarrow \frac{1}{((s-1)\cdot\mathrm{p}(s-1)))^{s-1}} \geq \frac{s\cdot\mathrm{p}(s)+1}{(s\cdot\mathrm{p}(s))^s} \tag{4.3}$$

$$\Leftrightarrow (s\cdot\mathrm{p}(s))^s \geq (s\cdot\mathrm{p}(s)+1)\,((s-1)\cdot\mathrm{p}(s-1)))^{s-1}. \tag{4.4}$$

Now, as $(s\cdot\mathrm{p}(s)+1) \leq 2s\cdot\mathrm{p}(s)$ it would suffice to show that $s\cdot\mathrm{p}(s) \geq 2^{(s-1)^{-1}}(s-1)\cdot\mathrm{p}(s-1)$. But this is clear:

$$2^{(s-1)^{-1}}(s-1)\cdot\mathrm{p}(s-1) \leq 2s\,\mathrm{p}(s-1) = 2s2^{(s-1)^2+3(s-1)} = 2s2^{s^2+s-2} \tag{4.5}$$

$$= s2^{s^2+s-1} \leq s2^{s^2+3s} = s\cdot\mathrm{p}(s). \tag{4.6}$$

So there exists a specific $(s-1)$-restriction τ where $\pi{\restriction}_\tau$ contains no bottlenecks. Therefore, by Lemma 3.1, there is a $\mathrm{Res}(s-1)$ refutation of size strictly less than

$$2^{c+2}L(s) = L(s-1).$$

in direct contradiction with our inductive assumption. ∎

Let us ponder what lower bound we have discovered. Due to the definition of $L(s)$ the proof can be carried as long as $n^{\frac{\delta}{\mathrm{d}(s)}}$ (where $\mathrm{d}(s) = (s\,\mathrm{p}(s))^s$ and $\mathrm{p}(s) = 2^{s^2+3s}$) is non-constant – indeed, growing in n – whereupon $n^{\frac{\delta(k-1)}{\mathrm{d}(s)}}$ grows significantly in k. This holds while $(s\,\mathrm{p}(s))^s \in o(\log n)$ which simplifies as

$$\log\log n \gg s\log(s\,\mathrm{p}(s)) = s\log(s2^{s^2+3s}) = s\log s + s^3 + 3s^2. \tag{4.7}$$

Clearly this holds if $s \in o((\log\log n)^{\frac{1}{3}})$. Hence we can deduce the following from Corollary 4.5.

Corollary 4.9. *Let $s \in o((\log\log n)^{\frac{1}{3}})$ and $k \le \log n$ be integers. Choose G so that* $\mathrm{Clique}(G, s, k)$ *holds (knowing that such exists). Then there are no Res(s) refutations of* $\text{Bin-Clique}^n_k(G)$ *of size smaller than $n^{\delta\frac{k-1}{\mathrm{d}(s)}}$, which is of the form $g(n)^k$ for some strictly increasing function g.*

Proof of Claim 1. Since $T_1, \dots, T_r$ are tuples in R, then $\Pr[\Sigma_R] \le \Pr[\Sigma_1 \wedge \dots \wedge \Sigma_r]$. Moreover $\Pr[\Sigma_1 \wedge \dots \wedge \Sigma_r] = \prod_{i=1}^r \Pr[\Sigma_i | \Sigma_1 \wedge \dots \wedge \Sigma_{i-1}]$. We will prove that for all $i = 1, \dots, r$,

$$\Pr[\Sigma_i | \Sigma_1 \wedge \dots \wedge \Sigma_{i-1}] \le \Pr[\Sigma_i]. \tag{4.8}$$

Hence the result follows from Lemma 4.10 which is proving that $\Pr[\Sigma_i] \le 1 - \frac{1}{\mathrm{p}(s)}$. By Lemma 4.11, to prove that Equation 4.8 holds, we show that

$$\Pr[\Sigma_i | \neg\Sigma_1 \vee \dots \vee \neg\Sigma_{i-1}] \ge \Pr[\Sigma_i]. \tag{4.9}$$

To prove Equation 4.9, let $B(T_j)$ be the set of blocks mentioned in T_j. If $B(T_i)$ and $B(T_1) \cup \dots \cup B(T_{i-1})$ are disjoint, then clearly $\Pr[\Sigma_i | \neg\Sigma_1 \vee \dots \vee \neg\Sigma_{i-1}] = \Pr[\Sigma_i]$. When $B(T_i)$ and $B(T_1) \cup \dots \cup B(T_{i-1})$ are not disjoint, we reason as follows: For each $\ell \in B(T_i)$, let T_i^ℓ be the set of variables in T_i mentioning block ℓ. T_i is hence partitioned into $\bigcup_{\ell \in B(T_i)} T_i^\ell$ and hence the event "T_i surviving σ", can be split into the independent events that T_i^ℓ survives σ, for $\ell \in B(T_i)$. Denote by Σ_i^ℓ the event "T_i^ℓ survives σ".

The following equalities hold:

$$\Pr[\neg\Sigma_i] = \Pr[\forall \ell \in B(T_i) : \neg\Sigma_i^\ell] = \prod_{\ell \in B(T_i)} \Pr[\neg\Sigma_i^\ell]$$

$$\Pr[\neg\Sigma_i | \neg\Sigma_1 \vee \dots \vee \neg\Sigma_{i-1}] = \prod_{\ell \in B(T_i)} \Pr[\neg\Sigma_i^\ell | \neg\Sigma_1 \vee \dots \vee \neg\Sigma_{i-1}]$$

Notice that T_i and $T_1, \dots, T_{i-1}$ are pairwise disjoint, hence knowing that some indices in blocks $\ell \in A$ are already chosen to kill some among $T_1, \dots, T_{i-1}$, only increases the chances that T_i survives (since fewer positions are left in the blocks $\ell \in A$ to potentially kill T_i). Thus

$$\Pr[\Sigma_i^\ell | \neg\Sigma_1 \vee \dots \vee \neg\Sigma_{i-1}] \ge \Pr[\Sigma_i^\ell]$$

whereupon

$$\Pr[\neg\Sigma_i^\ell | \neg\Sigma_1 \vee \ldots \vee \neg\Sigma_{i-1}] \leq \Pr[\neg\Sigma_i^\ell]$$

so we have

$$\begin{aligned}
\text{(4.10)}\quad \Pr[\Sigma_i | \neg\Sigma_1 \vee \ldots \vee \neg\Sigma_{i-1}] &= 1 - \prod_{\ell \in B(T_i)} \Pr[\neg\Sigma_i^\ell | \neg\Sigma_1 \vee \ldots \vee \neg\Sigma_{i-1}] \\
&\geq 1 - \prod_{\ell \in B(T_i)} \Pr[\neg\Sigma_i^\ell] \\
&= 1 - \Pr[\neg\Sigma_i] \\
&= \Pr[\Sigma_i].
\end{aligned}$$

This finally proves Equation 4.9. ◻

Lemma 4.10. *Let $s \in o((\log\log n)^{\frac{1}{3}})$ and let ρ be a random s-restriction. Then for all s-tuples S,*

$$\Pr[S \textit{ survives } \rho] \leq 1 - \frac{1}{\mathrm{p}(s)} = 1 - \frac{1}{2^{s^2+3s}}.$$

Proof. We prove that $\Pr[S$ does not survive $\rho] > \frac{1}{\mathrm{p}(s)}$. Let $\gamma = \frac{1}{2^{s+1}}$. For a block $i \in [k]$, let $S(i)$ be the set of literals of S in block i and let $r_i = |S(i)|$. Notice that $r_1 + \ldots + r_k = s$.

Since blocks are disjoint and since ρ acts independently on each block we have that

$$\text{(4.11)}\qquad \Pr[S \text{ does not survive } \rho] = \prod_{i=1}^{k} \Pr[S(i) \text{ does not survive } \rho].$$

For a generic block B with r distinct literals in S:

$$\text{(4.12)}\qquad \Pr[B \text{ does not survive } \rho] = \frac{\binom{\gamma\log n}{r}}{\binom{\log n}{r}} \cdot \frac{1}{2^r}$$

Expanding $\frac{\binom{\gamma\log n}{r}}{\binom{\log n}{r}}$ in Equation 4.12 we obtain

$$\frac{\gamma\log n \cdot (\gamma\log n\ - 1)\cdots(\gamma\log n\ - r + 1)}{\log n \cdot \log n\ - 1 \cdots \log n\ - r + 1} = \gamma\frac{\log n}{\log n} \cdot \gamma\frac{\log n\ - \frac{1}{\gamma}}{\log n\ - 1} \cdots \gamma\frac{\log n\ - \frac{r}{\gamma} + \frac{1}{\gamma}}{\log n\ - r + 1}.$$

Next, let us note that

$$1 = \frac{\log n}{\log n} > \frac{\log n\ - \frac{1}{\gamma}}{\log n\ - 1} > \cdots > \frac{\log n\ - \frac{r}{\gamma} + \frac{1}{\gamma}}{\log n\ - r + 1} > \frac{1}{2}$$

as long as $r \leq s$. This is because $2(\log n - 2^{s+1}s + 2^{s+1}) \geq \log n - s + 1$ reduces to $\log n \geq 2^{s+2}s - 2^{s+2} - s + 1$ which holds while $s \in o((\log\log n)^{\frac{1}{3}})$.

By Equation 4.11 and the previous discussion, $\Pr[B$ does not survive $\rho] > \frac{\gamma^r}{2^{2r}}$ and therefore

$$\Pr[S \text{ does not survive } \rho] > \prod_{i=1}^{k} \frac{\gamma^{r_i}}{2^{2r_i}} \tag{4.13}$$

$$= \frac{\gamma^{\sum_{i=1}^{k} r_i}}{2^{2(\sum_{i=1}^{k} r_i)}} \tag{4.14}$$

$$= \frac{\gamma^{s}}{2^{2s}} \tag{4.15}$$

The result follows since $\gamma = \frac{1}{2^{s+1}}$. ∎

Let us note that in Lemma 4.10 the probability that S survives ρ is maximised when $S = (\ell_{i_1,j_1}, \ldots, \ell_{i_s,j_s})$ is an s-tuple where all literals are bits from the same block.

LEMMA 4.11. *Let A, B, C be three events such that $\Pr[A], \Pr[B], \Pr[C] > 0$. If $\Pr[A|\neg B] \geq \Pr[A]$ then $\Pr[A|B] \leq \Pr[A]$.*

Proof. Consider the following steps:

$$\begin{array}{lcl} \Pr[A] & = & \Pr[A|B]\Pr[B] + \Pr[A|\neg B]\Pr[\neg B] \\ \Pr[A] & = & \Pr[A|B]\Pr[B] + \Pr[A|\neg B](1 - \Pr[B]) \\ \Pr[A] & \geq & \Pr[A|B]\Pr[B] + \Pr[A](1 - \Pr[B]) \\ \Pr[A]\Pr[B] & \geq & \Pr[A|B]\Pr[B] \\ \Pr[A] & \geq & \Pr[A|B] \end{array}$$

5. Res(s) and the weak Pigeonhole Principle.

For $n < m$, let Bin-PHP^m_n be the binary encoding of the (weak) Pigeonhole Principle. This involves variables $\omega_{i,j}$ that range over $i \in [m], j \in [\log n]$, where we assume for simplicity of our exposition that n is a power of 2. Its clauses are just

$$(\bigvee_{\ell=1}^{\log n} \omega_{i,\ell}^{1-a_\ell} \vee \bigvee_{\ell=1}^{\log n} \omega_{j,\ell}^{1-a_\ell}),$$

for $i \neq j$ and $a \in [n]$, where $\text{bin}(a)$ is $a_1 \ldots a_{\log n}$. For a comparison with the unary version see Section 9. First notice that an analog of Lemma 4.1 holds for the Pigeonhole Principle too.

LEMMA 5.1. *Suppose there are Resolution refutations of* PHP^m_n *of size S. Then there are* Res$(\log n)$ *refutations of* Bin-PHP^m_n *of size S.*

Let ρ be a partial assignment (a restriction) to the variables of Bin-PHP^m_n. We call ρ a *t-bit* restriction if ρ assigns t bits of each pigeon $b \in [m]$, i.e. t variables $\omega_{b,i}$ for each pigeon b. Let $v = (i, a)$ be an assignment meaning that pigeon i is assigned to hole a and let $a_1 \ldots a_{\log n}$ be the binary representation of a. We say that a restriction ρ is *consistent* with v if for all $j \in [\log n]$, $\sigma(\omega_{i,j})$ is either a_j or not assigned. We denote by Bin-PHP$^m_n\!\restriction_\rho$, Bin-PHP^m_n restricted by ρ. We will also consider the situation in which an s-bit restriction is applied to some Bin-PHP$^m_n\!\restriction_\rho$, creating Bin-PHP$^m_n\!\restriction_\tau$, where τ is an $s + t$-bit restriction.

Throughout this section, let $u = u(n,t) := 2((\log n) - t)$ and $u' := (\log n) - t$. We do not use these shorthands universally, but sometimes where otherwise the notation would look cluttered. We also occasionally write $(\log n) - t$ as $\log n\ - t$ (note the extra space). We say that a pigeon is *mentioned* in a CNF if some literal involving that pigeon appears in the CNF.

LEMMA 5.2. *Let ρ be a t-bit restriction for* Bin-PHP^m_n. *Any decision DAG for* $\text{Bin-PHP}^m_n\restriction_\rho$ *must contain a 1-CNF record which mentions* $\frac{n}{2^{t+1}}$ *pigeons.*

Proof. Let Adversary play in the following fashion. While some pigeon is not mentioned in the current record, let him give Prover a free choice to answer any one of its bits as true or false. Once a pigeon is mentioned once, then let Adversary choose a hole for that pigeon by choosing some assignment for the remaining unset bits (we will later need to prove this is always possible). Whenever another bit of an already mentioned pigeon is queried, then Adversary will answer consistently with the hole he has chosen for it. Only once all of a pigeon's bits are forgotten (not including those set by ρ), will Adversary forget the hole he assigned it.

It remains to argue that Adversary must force Prover to produce a 1-CNF record mentioning at least $\frac{n}{2^t}$ pigeons and for this it suffices to argue that Adversary can remain consistent with $\text{Bin-PHP}^m_n\restriction_\rho$ up until the point that such a 1-CNF record exists. For that it is enough to show that there is always a hole available for a pigeon for which Adversary gave its only currently questioned bit as a free choice (but for which ρ has already assigned some bits).

The current 1-CNF record is assumed to have fewer than $\frac{n}{2^t}$ literals and therefore must mention fewer than $\frac{n}{2^t}$ pigeons, each of which Adversary already assigned a hole. Each hitherto unmentioned pigeon that has just been given a free choice has $\log n\ - t - 1$ bits which corresponds to $\frac{n}{2^{t+1}}$ holes. Since we have assigned fewer than $\frac{n}{2^{t+1}}$ pigeons to holes, one of these must be available, and the result follows. ∎

Let $\xi(s)$ satisfy $\xi(1) = 1$ and $\xi(s) = \xi(s-1) + 1 + s$. Note that $\xi(s) = \Theta(s^2)$.

DEFINITION 5.3. *Let $s, t \geq 1$. We say that property* $\text{PHP}(s,t)$ *holds if for any t-bit restriction ρ to* Bin-PHP^m_n, *there are no* $\text{Res}(s)$ *refutations of* $\text{Bin-PHP}^m_n\restriction_\rho$ *of size smaller than* $e^{\frac{n}{4^{\xi(s)+1}s!2^t u^{\xi(s)}}} = \exp(\frac{n}{4^{\xi(s)+1}s!2^t u^{\xi(s)}})$.

THEOREM 5.4. *Let ρ be a t-bit restriction for* Bin-PHP^m_n. *Any decision DAG for* $\text{Bin-PHP}^m_n\restriction_\rho$ *is of size* $\geq e^{\frac{n}{2^{t+2}u}}$ *(which is $2^{\Omega(\frac{n}{\log n})}$ at $t = 0$).*

Proof. Call a *bottleneck* a 1-CNF record in the decision DAG that mentions $\frac{n}{2^{t+2}}$ pigeons. Now consider a random restriction that picks for each pigeon one bit uniformly at random and sets this to 0 or 1 with equal probability. The probability that a bottleneck survives (is not falsified by) the random restriction is no more than

$$\left(\frac{u'-1}{u'} + \frac{1}{2u'}\right)^{\frac{n}{2^{t+2}}} = \left(1 - \frac{1}{2u'}\right)^{u' \cdot \frac{n}{2^{t+2}u'}} = \left(1 - \frac{1}{u}\right)^{u \cdot \frac{n}{2^{t+2}u}} \leq \frac{1}{e^{\frac{n}{2^{t+2}u}}},$$

since $e^{-x} = \lim_{m\to\infty}(1 - x/m)^m$ and indeed $e^{-x} \geq (1 - x/m)^m$ when $x, m \geq 1$.

Now suppose for contradiction that we have fewer than $e^{\frac{n}{2^{t+2}u}}$ bottlenecks in a decision DAG for $\text{Bin-PHP}^m_n\restriction_\rho$. By the union bound there is a random restriction that kills all bottlenecks and this leaves a decision DAG for some $\text{Bin-PHP}^m_n\restriction_\sigma$, where σ is a $(t+1)$-bit restriction for Bin-PHP^m_n. However, we know from Lemma 5.2 that such a refutation must involve a 1-CNF record mentioning $\frac{n}{2^{t+2}}$ pigeons. This is now the desired contradiction. ∎

While m is linear in n, the previous theorem could have been proved, like Lemma 4.7, by the size-width trade-off. However, the method of random restrictions used here could not be easily applied there, due to the randomness of G.

COROLLARY 5.5. *Property* $\text{PHP}(1,t)$ *holds, for each $t < \log n$.*

Note that, PHP$(1,t)$ yields only trivial bounds as t approaches $\log n$.

The random restrictions that we use in this section will be quite different from those used in the previous section (Section 4). Indeed, they will be much simpler. A *random s-bit restriction* is simply an assignment uniformly at random to some s unassigned bits of each pigeon, where this subset of s bits was itself picked uniformly at random. Note that we already used a random 1-bit restriction in the proof of Theorem 5.4.

LEMMA 5.6. *Let s be an integer, $s \geq 1$ and $s+t < \log n$. Let σ be a random s-bit restriction over* Bin-PHP$^m_n\restriction_\rho$ *where ρ is itself some t-bit restriction over* Bin-PHP^m_n. *Then for all s-tuples S,*

$$\Pr[S \textit{ survives } \sigma] \leq 1 - \frac{1}{u^s}$$

.

Proof. The proof is analogous to Lemma 4.10. We prove that $\Pr[S \text{ does not survive } \sigma] \geq \frac{1}{u^s}$. For a pigeon $i \in [m]$, let $S(i)$ be the bits of pigeon i mentioned in literals of S. Let $r_i = |S(i)|$. Hence $\sum_{i \in [m]} r_i = s$. Since σ acts independently on each pigeon remaining after ρ is applied to Bin-PHP^m_n,

$$\Pr[S \text{ does not survive } \sigma] = \prod_{i=1}^{m} \Pr[S(i) \text{ does not survive } \sigma]. \tag{5.1}$$

Now, similarly for the case of blocks in Lemma 4.10, for each $S(i)$ we have that:

$$\Pr[S(i) \text{ does not survive } \sigma] = \frac{s}{\log n\ - t} \cdot \frac{s-1}{\log n\ - t - 1} \cdots \frac{s - r_i + 1}{\log n - t - r_i + 1} \cdot \frac{1}{2^{r_i}}.$$

Now, $\frac{s-r_i+1}{\log n - t - r_i + 1} = \frac{s - r_i + 1}{u' - r_i + 1} > \frac{1}{u'}$ since $s \geq r_i$, $u' > s$ and $s > 1$. Hence,

$$\frac{s}{\log n\ - t} \cdot \frac{s-1}{\log n\ - t - 1} \cdots \frac{s - r_i + 1}{\log n - t - r_i + 1} \cdot \frac{1}{2^{r_i}} > \frac{1}{(2u')^{r_i}} = \frac{1}{u^{r_i}}$$

The claim immediately follows by Equation 5.1 and the fact that $\sum_{i \in [m]} r_i = s$. ∎

Let us note that in Lemma 5.6 the probability that S survives ρ is maximised when $S = (\ell_{i_1,j_1}, \ldots, \ell_{i_s,j_s})$ is an s-tuple where all literals are from different pigeons. This is essentially the opposite case from Lemma 4.10 and demonstrates how our random restrictions are different between the two cases.

THEOREM 5.7. *Let $s > 1$ and $s + t < \log n$. Then,* PHP$(s-1, s+t)$ *implies* PHP(s,t).

Proof. We proceed by contraposition. Assume there is some t-bit restriction ρ so that there exists a Res(s) refutation π of Bin-PHP$^m_n\restriction_\rho$ with size less than $e^{\frac{n}{4^{\xi(s)+1} \cdot s! 2^t u^{\xi(s)}}} = \exp(\frac{n}{4^{\xi(s)+1} \cdot s! 2^t u^{\xi(s)}})$.

Call a *bottleneck* a CNF record that has covering number $\geq \frac{n}{4^{\xi(s)} \cdot (s-1)! 2^t u^{\xi(s-1)}}$. In such a CNF record, by dividing by s and u, it is always possible to find $r := \frac{n}{4^{\xi(s)} s! 2^t u^{\xi(s-1)+1}}$ s-tuples of literals $(\ell^1_1, \ldots, \ell^s_1), \ldots, (\ell^1_r, \ldots, \ell^s_r)$ so that each s-tuple is a clause in the CNF record and no pigeon appearing in the ith s-tuple also appears in the jth s-tuple (when $i \neq j$). This important independence condition plays a key role. Now consider a random restriction that, for each pigeon, picks uniformly at random s bit positions and sets these to 0 or 1 with equal probability. The probability that the ith of the r s-tuples survives the restriction is maximised when each each variable among the s describes a different pigeon (by Lemma 5.6) and is therefore bounded above by

$$\left(1 - \frac{1}{u^s}\right)$$

whereupon

$$\left(1 - \frac{1}{u^s}\right)^{\frac{n}{4^{\xi(s)} s! 2^t u^{\xi(s-1)+1}}} = \left(1 - \frac{1}{u^s}\right)^{\frac{n u^s}{4^{\xi(s)} s! 2^t u^{(\xi(s-1)+1+s)}}}$$

which is $\leq 1/e^{\frac{n}{4^{\xi(s)} s! \cdot 2^t u^{\xi(s)}}} \leq 1/e^{\frac{n}{4^{\xi(s)+1} s! \cdot 2^t u^{\xi(s)}}}$. Supposing therefore that there are fewer than $e^{\frac{n}{4^{\xi(s)+1} s! \cdot 2^t u^{\xi(s)}}}$ bottlenecks, one can deduce a random restriction that kills all bottlenecks. What remains after doing this is a $\mathrm{Res}(s)$ refutation of some $\text{Bin-PHP}^m_n\restriction_\sigma$, where σ is a $s+t$-bit restriction, which moreover has covering number $< \frac{n}{4^{\xi(s)} \cdot (s-1)! 2^t u^{\xi(s-1)}}$. But if the remaining $\mathrm{Res}(s)$ refutation is of size $< e^{\frac{n}{4^{\xi(s)+1} s! \cdot 2^t u^{\xi(s)}}}$ then, from Lemma 3.1, it would give a $\mathrm{Res}(s-1)$ refutation of size

$$< e^{\frac{n}{4^{\xi(s)} \cdot (s-1)! 2^t u^{\xi(s-1)}}} \cdot e^{\frac{n}{4^{\xi(s)+1} s! \cdot 2^t u^{\xi(s)}}} = e^{\frac{n}{4^{\xi(s)} \cdot (s-1)! 2^t u^{\xi(s-1)}} (1 + \frac{1}{4 s u^{s+1}})}$$

$$< e^{\frac{2n}{4^{\xi(s)} \cdot (s-1)! 2^t u^{\xi(s-1)}}} < e^{\frac{n}{4^{\xi(s)} \cdot (s-1)! 2^{t-1} u^{\xi(s-1)}}} < e^{\frac{n}{4^{\xi(s)-s} \cdot (s-1)! 2^{s+t} u^{\xi(s-1)}}},$$

since $4^s > 2^{s+1}$, which equals $e^{\frac{n}{4^{\xi(s-1)+1} \cdot (s-1)! 2^{s+t} u^{\xi(s-1)}}}$ in contradiction to the inductive hypothesis. ∎

THEOREM 5.8. *Fix $\lambda, \mu > 0$. Any refutation of* Bin-PHP^m_n *in* $\mathrm{Res}(\sqrt{2} \log^{\frac{1}{2}-\lambda} n)$ *is of size* $2^{\Omega(n^{1-\mu})}$.

Proof. First, let us claim that $\mathrm{PHP}(\sqrt{2} \log^{\frac{1}{2}-\lambda} n, 0)$ holds (and this would hold also at $\lambda = 0$). Let $s = \sqrt{2} \log^{\frac{1}{2}-\lambda} n$. Then we use Corollary 5.5 at $t = \frac{s(s+1)}{2}$ before applying Theorem 5.7 repeatedly to obtain our answer. Noting $\frac{s(s+1)}{2} < \log n$, the claim follows.

Now let us look at the bound we obtain by plugging in to $e^{\frac{n}{4^{\xi(s)+1} \cdot s! 2^t u^{\xi(s)}}}$ at $s = \sqrt{2} \log^{\frac{1}{2}-\lambda} n$ and $t = 0$. We recall $\xi(s) = \Theta(s^2)$. Note that, when $\lambda > 0$, each of $4^{\xi(s)+1}$, $s!$ and $\log^{\xi(s)} n$ is $o(n^\mu)$. The result follows. ∎

5.1. The treelike case. Concerning the Pigeonhole Principle, we can prove that the relationship between PHP^{n+1}_n and Bin-PHP^{n+1}_n is different for treelike Resolution from general Resolution. In particular, for very weak Pigeonhole Principles, we know the binary encoding is harder to refute in general Resolution; whereas for treelike Resolution it is the unary encoding which is the harder.

THEOREM 5.9. *The treelike Resolution complexity of* Bin-PHP^m_n *is* $2^{\Theta(n)}$.

Proof. For the lower bound, one can follow the proof of Lemma 5.2 with $t = 0$ and find n free choices on each branch of the tree. Following the method of Riis [64], we uncover a subtree of the decision tree of size 2^n.

For an upper bound of 2^{2n} we pursue the following strategy. First we choose some $n+1$ pigeons to question. We then question all of them on their first bit and separate these into two sets T_1 and F_1 according to whether this was answered true or false. If n is a power of 2, choose the larger of these two sets (if they are the same size then choose either). If n is not a power of two, the matter is mildly complicated, and one must look at how many holes are available with the first bit set to 1, say h_1^1; versus 0, say h_1^0. At least one of $|T_1| > h_1^1$ or $|F_1| > h_1^0$ must hold and one can choose between T_1 and F_1 correspondingly. Now question the second bit, producing two sets T_2 and F_2, and iterate this argument. We will reach a contradiction in $\log n$ iterations since we always choose a set of maximal size. The depth of our tree is bounded above by $(n+1) + (\frac{n}{2}+1) + (\frac{n}{4}+1) + \cdots = 2n + \log n$ and the result follows. ◻

6. The SA size lower bound for the binary Pigeonhole Principle. In this section we study the inequalities derived from the binary encoding of the Pigeonhole principle, whose axioms we remind the reader of now. Bin-PHP^m_n has, for each two distinct pigeons $i \neq i' \in [m]$ and each hole $a \in [n]$, the axiom $\sum_{j=1}^{\log n} \omega_{i,j}^{(1-a_j)} + \sum_{j=1}^{\log n} \omega_{i',j}^{(1-a_j)} \geq 1$, where $a_1 \dots a_{\log n}$ is the binary representation of a. We first prove a certain SA rank lower bound for a version of the binary PHP, in which only a subset of the holes is available.

Lemma 6.1. *Let $H \subseteq [n]$ be a subset of the holes and let us consider* Bin-PHP$^m_{|H|}$ *where each pigeon can go to a hole in H only. Any* SA *refutation of* Bin-PHP$^m_{|H|}$ *involves a term that mentions at least $|H|$ pigeons.*

Proof. We get a valuation v from a partial matching in an obvious way. That is, if a pigeon i is assigned to hole a, whose representation in binary is $a_1 \dots a_{\log n}$, then we set each $\omega_{i,j}^{a_j}$ to a_j. We say that a product term $P = \prod_{j \in J} \omega_{i_j,k_j}^{b_j}$ mentions the set of pigeons $M = \{i_j : j \in J\}$. Let us denote the number of available holes by $n' := |H|$. Every product term that mentions at most n' pigeons is assigned a value $v(P)$ as follows. The set of pigeons mentioned in M is first extended arbitrarily to a set M' of exactly n' pigeons. $v(P)$ is then the probability that a matching between M' and H taken uniformly at random is consistent with the product term P. In other words, $v(P)$ is the number of perfect matchings between M' and H that are consistent with P, divided by the total, $(n')!$. Obviously, this value does not depend on how M is extended to M'. Also, it is symmetric, i.e. if π is a permutation of the pigeons, $v\left(\prod \omega_{i_j,k_j}^{b_j}\right) = v\left(\prod \omega_{\pi(i_j),k_j}^{b_j}\right)$.

All lifts of axioms of equality $\omega_{j,k} + \neg\omega_{j,k} = 1$ are automatically satisfied since a matching consistent with P is consistent either with $P\omega_{j,k}^b$ or with $P\omega_{j,k}^{1-b}$ but not with both, and thus

$$v(P) = v\left(P\omega_{j,k}^b\right) + v\left(P\omega_{j,k}^{1-b}\right).$$

Regarding the lifts of the disequality of two pigeons $i \neq j$ in one hole, that is, the inequalities coming from the only clauses in Bin-PHP$^m_{|H|}$, it is enough to observe that it is consistent with any perfect matching, i.e. at least one variable on the LHS is one under such a matching. Thus, for a product term P, any perfect matching consistent with P will also be consistent with $P\omega_{i,k}^{1-b_k}$ or with $P\omega_{j,k}^{1-b_k}$ for some k. ◻

6.1. The ordinary Pigeonhole Principle. The proof of the size lower bound for the Bin-PHP$^{n+1}_n$ is then, by a standard random restriction argument, combined with the rank lower bound above. Assume, without loss of generality, that n is a power of two. For the random restrictions $\mathcal{R}$, we consider the pigeons one by one and with probability $1/4$ we assign the pigeon uniformly at random to one of the holes still available. We first need to show that the restriction is "good" with high probability, i.e. neither too big nor too small. The former is needed so that in the restricted version we have a good lower bound, while the latter will be needed to show that a good restriction coincides well with any reasonably big term, in the sense that they have in common a sufficiency of pigeons.

We will make use of the following version of the Chernoff Bound as appears in [55].

Lemma 6.2 (Theorems 4.4 and 4.5 in [55]). *Let $X_1, X_2, \ldots, X_n$ be independent 0/1 random variables with $Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = E[X]$. Then, for every δ, $0 < \delta \leq 1$, the following bound holds*

$$Pr[X \geq (1+\delta)\mu] \leq e^{\frac{-\mu\delta^2}{3}}$$

and similarly

$$Pr[X \leq (1-\delta)\mu] \leq e^{\frac{-\mu\delta^2}{3}}.$$

Lemma 6.3. *If $|\mathcal{R}|$ is the number of pigeons (or holes) assigned by $\mathcal{R}$, the probability that $|\mathcal{R}| > \frac{3(n+1)}{8}$ is at most $e^{-\frac{(n+1)}{48}}$.*

Proof. We use the Chernoff Bound from Lemma 6.2. We have $p_i = \frac{1}{4}$ (and thus $\mu = \frac{n+1}{4}$) and $\delta = \frac{1}{2}$. Thus, the probability the restriction assigns more than $\frac{3(n+1)}{8}$ pigeons to holes is at most $e^{-(n+1)/48}$. ∎

We first prove that any given wide product term, i.e. a term that mentions a constant fraction of the pigeons, survives the random restrictions with exponentially small probability.

Lemma 6.4. *Let P be a product term that mentions at least $\frac{n+1}{2}$ pigeons. The probability that P does not evaluate to zero under the random restrictions is at most $\left(\frac{5}{6}\right)^{n/16}$ (for n large enough).*

Proof. We will desire $|\mathcal{R}| \leq \frac{3(n+1)}{8}$ to ensure that at least $\frac{5(n+1)}{8}$ holes remain unused in $\mathcal{R}$ (for n large enough). This will involve the probability $e^{-(n+1)/48}$ from Lemma 6.3.

A further application of the Chernoff Bound from Lemma 6.2 ($\mu = \frac{n+1}{8}$, $\delta = -\frac{1}{2}$) gives the probability that fewer than $\frac{n+1}{16}$ pigeons mentioned by P are assigned by $\mathcal{R}$ is at most $e^{-(n+1)/96}$.

For each of these assigned pigeons the probability that a single bit-variable in P belonging to the pigeon is set by $\mathcal{R}$ to zero is at least $\frac{1}{5}$. This is because when $\mathcal{R}$ sets the pigeon, and thus the bit-variable, there were at least $\frac{5(n+1)}{8}$ holes available, while at most $\frac{n+1}{2}$ choices set the bit-variable to one. The difference – which will be a lower bound on the number of holes available setting the selected bit to 0 – is $\frac{n+1}{8}$ which when divided by $\frac{5(n+1)}{8}$ (to normalise the probability) gives $\frac{1}{5}$. Thus P survives under $\mathcal{R}$ with probability at most $e^{-(n+1)/48} + e^{-(n+1)/96} + \left(\frac{4}{5}\right)^{(n+1)/16} < \left(\frac{5}{6}\right)^{n/16}$. ∎

Finally, we can prove that

Theorem 6.5. *Any* SA *refutation of the* Bin-PHP$_n^{n+1}$ *has to contain at least $\left(\frac{7}{6}\right)^{n/16}$ terms.*

Proof. Assume for a contradiction, that there is a smaller refutation. We wish to argue that there is a random restriction with $|\mathcal{R}| \leq \frac{3(n+1)}{8}$ that evaluates to zero all terms that mention at least $\frac{n+1}{2}$ pigeons. There are at most $\left(\frac{7}{6}\right)^{n/16}$ such terms so an application of the union bound together with Lemma 6.3 and Lemma 6.4 gives a probability that some term mentioning at least $\frac{n+1}{2}$ pigeons does not evaluate to zero of

$$\left(\frac{5}{6}\right)^{n/16} \times \left(\frac{7}{6}\right)^{n/16} + e^{-(n+1)/48} < 1.$$

Now we apply the random restriction which we know must exist to leave no terms mentioning at least $\frac{n+1}{2}$ pigeons in an SA refutation of the binary $\mathrm{PHP}^{m'}_{n'}$, where $m' > n' \geq \frac{5(n+1)}{8}$. However, since $n' > \frac{n+1}{2}$, this contradicts Lemma 6.1. ∎

COROLLARY 6.6. *Any* SA *refutation of the* Bin-PHP^{n+1}_{n} *must have size* $2^{\Theta(n)}$.

Proof. The size lower bound comes from the previous theorem. We know that there is a $2^{\Theta(n)}$ upper bound in treelike Resolution from Theorem 5.9 and the result follows from the standard simulation of Resolution by SA which increases refutations by no more than a factor which is a polynomial in n [29]. ∎

6.2. The weak Pigeonhole Principle. We now consider the so-called weak binary PHP, Bin-PHP^{m}_{n}, where m is potentially much larger than n. The weak unary PHP^{m}_{n} is interesting because it admits (significantly) subexponential-in-n refutations in Resolution when m is sufficiently large [24]. It follows that this size upper bound is mirrored in SA. However, as proved earlier in this article the weak binary Bin-PHP^{m}_{n} remains almost-exponential-in-n for minimal refutations in Resolution. We will see here that the weak binary Bin-PHP^{m}_{n} remains almost-exponential-in-n for minimally sized refutations in SA. In this weak binary case, the random restrictions $\mathcal{R}$ above do not work, so we apply quite different restrictions $\mathcal{R}'$ that are as follows: for each pigeon select independently a single bit uniformly at random and set it to 0 or 1 with probability of 1/2 each.

We can easily prove the following

LEMMA 6.7. *A product term P that mentions n' pigeons does not evaluate to zero under $\mathcal{R}'$ with probability at most $e^{-n'/2\log n}$.*

Proof. For each pigeon mentioned in P, the probability that the bit-variable present in P is set by the random restriction is $\frac{1}{\log n}$, and if so, the probability that the bit-variable evaluates to zero is $\frac{1}{2}$. Since this happens independently for all n' mentioned pigeons, the probability that they all survive is at most $\left(1 - \frac{1}{2\log n}\right)^{n'}$. ∎

LEMMA 6.8. *The probability that $\mathcal{R}'$ fails to have, for each $k \in [\log n]$ and $b \in \{0,1\}$, at least $\frac{m}{4\log n}$ pigeons with the kth bit set to b, is at most $e^{-n/48\log n}$.*

Proof. We apply the Chernoff Bound of Lemma 6.2 to deduce that for each bit position k, $1 \leq k \leq (\log n)$ and a value b, 0 or 1, the probability that there are fewer than $\frac{m}{4\log n}$ pigeons for which the kth bit is set to b is at most $e^{-m/24\log n}$. This uses $\mu = \frac{m}{2\log n}$ and $\delta = -\frac{1}{2}$. Since $m > n$, by the union bound, the probability that this holds for some position k and some value b is at most $(2\log n)e^{-m/24\log n} \leq e^{-n/48\log n}$. ∎

In order to conclude our result, we will profit from a graph-theoretic treatment of Hall's Marriage Theorem [39]. Suppose G is a finite bipartite graph with bipartitions X and Y, then an X-saturating matching is a matching which covers every vertex in X. For a subset W of X, let $N_G(W)$ denote the neighborhood of W in G, i.e. the set of all vertices in Y adjacent to some element of W.

THEOREM 6.9 ([39] (see Theorem 5.1 in [69])). *Let G be a finite bipartite graph with bipartitions X and Y. There is an X-saturating matching if and only if for every* *subset W of X, $|W| \leq |N_G(W)|$.*

Corollary 6.10. *Any* SA *refutation of the* Bin-PHP^m_n*, $m > n$, has to contain at least $e^{n/32\log^2 n}$ terms.*

Proof. Assume for a contradiction, that there is a refutation with fewer than $e^{n/32\log^2 n}$ product terms. We want to argue that there is a random restriction that evaluates all terms that mention at least $\frac{n}{4\log n}$ pigeons to zero while satisfying the condition of Lemma 6.8. Using a union bound and Lemma 6.7 we upper bound the probability this fails to happen as $e^{-n/8\log^2 n} \cdot e^{n/32\log^2 n} + e^{-n/48\log n} < 1$ so such a random restriction $\mathcal{R}'$ does exist.

Then, $\mathcal{R}'$ leaves at least $\frac{m}{4\log n}$ pigeons of each type (k, b), i.e. the kth bit of the pigeon is set to b. Recalling $m \geq n$, we now pick a set of pigeons S that has $(*)$ precisely $\frac{n}{4\log n}$ pigeons of each type and thus is of size $n/2$.

We will give an evaluation of the restricted principle which contradicts that the original object was a refutation. This new principle is not a copy of the weak Pigeonhole Principle it is rather a distorted variant thereof. We evaluate any product term P that mentions at most $\frac{n}{4\log n}$ pigeons by first relabeling the mentioned pigeons, injectively, using the labels of pigeons in S while preserving types, which we can do due to property $(*)$, and then giving it a value as before. That is, by taking the probability that a perfect matching between S and some set of $n/2$ holes consistent with the random restriction, is consistent with P. Indeed, we take the average here for all possibilities of the set of $n/2$ holes consistent with the random restriction. In this fashion, the valuation is clearly dependent on the random restriction.

To finish the proof, we need to show that such a set of $n/2$ holes exists, that is, such a matching exists. But this follows trivially from Theorem 6.9 as every pigeon has $n/2$ holes available, so at least the same applies to any set of pigeons. ∎

7. The SA rank upper bound for Ordering Principle with equality. Let us remind ourselves of the Ordering Principle in both unary and binary. Its negation can be expressed in first-order logic as:

$$\forall x, y, z \exists w \; \neg R(x, x) \wedge (R(x, y) \wedge R(y, z) \rightarrow R(x, z)) \wedge R(x, w).$$

Its usual unary and binary encodings, à la Riis, may be given as follows:

$$\begin{array}{cc|cc}
\multicolumn{2}{c}{\mathrm{OP}_n : \underline{\textit{Unary encoding}}} & \multicolumn{2}{c}{\text{Bin-OP}_n : \underline{\textit{Binary encoding}}} \\
\neg v_{i,i} & \forall i \in [n] & \neg \nu_{i,i} & \forall i \in [n] \\
\neg v_{i,j} \vee \neg v_{j,k} \vee v_{i,k} & \forall i, j, k \in [n] & \neg \nu_{i,j} \vee \neg \nu_{j,k} \vee \nu_{i,k} & \forall i, j, k \in [n] \\
\neg w_{i,j} \vee v_{i,j} & \forall i, j \in [n] & \bigvee_{i \in [\log n]} \omega_{i,j}^{1-a_i} \vee \nu_{j,a} & \forall j, a \in [n] \\
\bigvee_{i \in [n]} w_{i,j} & \forall j \in [n] & \multicolumn{2}{c}{\text{where } a_1 \dots a_{\log n} = \mathrm{bin}(a)}
\end{array}$$

Note that we placed the witness in the variables $w_{i,x}$ as the first argument and not the second, as we had in the introduction. This is to be consistent with the $v_{i,j}$ and the standard formulation of OP as the least, and not greatest, number principle. A more traditional form of the (unary encoding of the) OP$_n$ has clauses $\bigvee_{i \in [n]} v_{i,j}$ which are consequent on $\bigvee_{i \in [n]} w_{i,j}$ and $\neg w_{i,j} \vee v_{i,j}$ (for all $i \in [n]$).

In SA, we wish to discuss the encoding of the Ordering Principle (and Pigeonhole Principle) as ILPs *with equality.* For this, we take the unary encoding but instead of translating the wide clauses (e.g. from the OP) from $\bigvee_{i\in[n]} w_{i,x}$ to $w_{1,x}+\ldots+w_{n,x}\geq 1$, we instead use $w_{1,x}+\ldots+w_{n,x}=1$. This makes the constraint at-least-one into exactly-one (which is a priori enforced in the binary encoding). A reader favouring a specific example may consider the Ordering Principle as the combinatorial principle of the following lemma.

Lemma 7.1. *Let* C *be any combinatorial principle expressible as a first order formula in* Π_2*-form with no finite models. Suppose the unary encoding of* C *with equalities has an* SA *refutation of rank* r *and size* s*. Then the binary encoding of* C *has an* SA *refutation of rank at most* $r\log n$ *and size at most* s*.*

Proof. We take the SA refutation of the unary encoding of C with equalities of rank r, in the form of a set of inequalities, and build an SA refutation of the binary encoding of C of rank $r\log n$, by substituting terms $w_{x,a}$ in the former with $\omega_{x,1}^{a_1}\cdots\omega_{x,\log n}^{a_{\log n}}$, where $a_1\ldots a_{\log n}=\text{bin}(a)$, in the latter. $\neg w_{x,a}$ is substituted by $1-\omega_{x,1}^{a_1}\cdots\omega_{x,\log n}^{a_{\log n}}$. Variables $v_{x,a}$ and $\neg v_{x,a}$ are substituted by $\nu_{x,a}$ and $1-\nu_{x,a}$, respectively.

It remains to show we can build the translation of the SA with equalities axioms in the binary case from the true axioms of the binary case. Axioms from the binary case that involve only variables ν_{x_a} appear perfectly reproduced. Axioms of the form

$$\sum_{a\in[n]:a_1\ldots a_{\log n}=\text{bin}(a)} \omega_{x,1}^{a_1}\cdots\omega_{x,\log n}^{a_{\log n}}=1$$

follow from the equalities (3.5). Finally, axioms of the form $\omega_{x,1}^{a_1}\ldots\omega_{x,\log n}^{a_{\log n}}\leq \nu_{x,a}$, can also be built since $\omega_{x,j}\overline{\omega}_{x,j}=0$ for each $j\in[\log n]$. Let us explain this in detail. The axioms are of the form $\bigvee_{i\in[\log n]}\omega_{j,i}^{1-a_i}\vee\nu_{j,a}$ which becomes $\omega_{j,1}^{1-a_1}+\cdots+\omega_{j,\log n}^{1-a_{\log n}}+\nu_{j,a}\geq 1$. We now lift through by $\omega_{j,1}^{a_1},\ \ldots,\ \omega_{j,\log n}^{a_{\log n}}$ to obtain $\omega_{x,1}^{a_1}\ldots\omega_{x,\log n}^{a_{\log n}}\leq\omega_{x,1}^{a_1}\ldots\omega_{x,\log n}^{a_{\log n}}\nu_{x,a}\leq\nu_{x,a}$. ◻

The unary *Ordering Principle* (OP_n) *with equality* has the following set of SA axioms:

$$\begin{aligned} \mathit{self}: v_{i,i}=0 \quad \forall\, i\in n\\ \mathit{trans}: v_{i,k}-v_{i,j}-v_{j,k}+1\geq 0 \quad \forall\, i,j,k\in[n]\\ \mathit{impl}: v_{i,j}-w_{i,j}\geq 0 \quad \forall\, i,j\in[n]\\ \mathit{lower}: \textstyle\sum_{i\in[n]} w_{i,j}-1=0 \quad \forall\, j\in[n]\end{aligned}$$

Note that we need the w-variables since we use the equality form. Axioms of the form $\sum_{i\in[n]} x_{i,j}-1=0$ made just from v-variables are plainly incompatible with, e.g., transitivity. Strictly speaking Sherali-Adams is defined for inequalities only. An equality axiom $a=0$ is simulated by the two inequalities $a\geq 0, -a\geq 0$, which we refer to as the *positive* and *negative* instances of that axiom, respectively. Also, note that we have used $v_{i,j}+\overline{v}_{i,j}=1$ to derive this formulation. We call two product terms *isomorphic* if one product term can be gotten from the other by relabelling the indices appearing in the subscripts by a permutation.

Theorem 7.2. *The* SA *rank of the* OP_n *with equality is at most* 2 *and* SA *size at most polynomial in* n*.*

Proof. Note that if the polytope $\mathcal{P}_2^{\text{OP}_n}$ is nonempty there must exist a point where any isomorphic variables are given the same value. We can find such a point by averaging an asymmetric valuation over all permutations of $[n]$.

So suppose towards a contradiction there is such a symmetric point. First note $v_{i,i}=$

$w_{i,i} = 0$ by *self* and *impl*. We start by lifting the jth instance of *lower* by $v_{i,j}$ to get

$$w_{i,j}v_{i,j} + \sum_{k \neq i,j} w_{k,j}v_{i,j} = v_{i,j}.$$

Equating (by symmetry with respect to k) the product terms $w_{k,j}v_{i,j}$ this is actually

$$w_{i,j}v_{i,j} + (n-2)w_{k,j}v_{i,j} = v_{i,j}.$$

Lift this by $w_{k,j}$ to get

$$w_{k,j}w_{i,j}v_{i,j} + (n-2)w_{k,j}v_{i,j} = w_{k,j}v_{i,j}.$$

We can delete the leftmost product term by proving it must be 0. Let us take an instance of *lower* lifted by $w_{k,j}v_{i,j}$ for any $k \neq i, j$ along with an instance of monotonicity $w_{k,j}w_{m,j}v_{i,j} \geq 0$ for every $m \neq j, k$:

$$\begin{aligned} & w_{k,j}v_{i,j}\left(1 - \sum_{m \neq j} w_{m,j}\right) + \sum_{m \neq j,k,i} w_{k,j}w_{m,j}v_{i,j} \\ &= -\sum_{m \neq k,j} w_{k,j}w_{m,j}v_{i,j} + \sum_{m \neq j,k,i} w_{k,j}w_{m,j}v_{i,j} \\ &= -w_{k,j}w_{i,j}v_{i,j}. \end{aligned} \tag{7.1}$$

The left hand side of this equation is greater than 0 so we can deduce $w_{k,j}w_{i,j}v_{i,j} = 0$.

This results in

$$(n-2)w_{k,j}v_{i,j} = w_{k,j}v_{i,j} \quad \text{which is} \quad w_{k,j}v_{i,j} = 0.$$

We lift *impl* by $w_{i,j}$ to obtain $w_{i,j} \leq w_{i,j}v_{i,j}$. Monotonicity gives us the opposite inequality and we can proceed as if we had the equality $w_{k,j}v_{k,j} = w_{k,j}$ (as we are using equality as shorthand for inequality in both directions) .
So repeating the derivation of $w_{k,j}v_{i,j} = 0$ for every $i \neq k$ and then adding $w_{k,j}v_{k,j} = w_{k,j}$ gets us $\sum_m w_{k,j}v_{m,j} = w_{k,j}$. Repeating this again for every k and summing up gives

$$0 = \sum_{k,m} w_{k,j}v_{m,j} - \sum_k w_{k,j} = \sum_{k,m} w_{k,j}v_{m,j} - 1$$

with the last equality coming from the addition of the positive *lower* instance $\sum_k w_{k,j} - 1 = 0$. Finally adding the lifted *lower* instance $v_{m,j} - \sum_k w_{k,j}v_{m,j} = 0$ for every m gives

$$\sum_m v_{m,j} = 1. \tag{7.2}$$

By lifting the *trans* axiom $v_{i,k} - v_{i,j} - v_{j,k} + 1 \geq 0$ by $v_{j,k}$ we get

$$v_{i,k}v_{j,k} - v_{i,j}v_{j,k} \geq 0. \tag{7.3}$$

Now, due to a manipulation similar to Equation (7.1) using Equation (7.2)

$$
\begin{aligned}
& v_{k,j}v_{i,j}\left(1-\sum_{m\neq j}v_{m,j}\right)+\sum_{m\neq j,k,i}v_{k,j}v_{m,j}v_{i,j} \\
&= -\sum_{m\neq k,j}v_{k,j}v_{m,j}v_{i,j}+\sum_{m\neq j,k,i}v_{k,j}v_{m,j}v_{i,j} \\
&= -v_{k,j}v_{i,j}v_{i,j} \qquad (7.4) \\
&= -v_{k,j}v_{i,j}. \qquad (7.5)
\end{aligned}
$$

Thus, $v_{i,k}v_{j,k}$ must be zero whenever $i \neq j$. Along with Equation (7.3) we derive $v_{i,j}v_{j,k} = 0$. Noting $v_{i,j}v_{j,i} = 0$ follows from *trans* and *self*, we lift Equation (7.2) by $v_{j,x}$ for some x to get

$$v_{j,x}\sum_{m}v_{m,j} = \sum_{m\neq x,j}v_{m,j}v_{j,x} = v_{j,x}$$

where we know the left hand side is zero (Equation (7.3)). Thus we can derive $v_{i,j} = 0$ for any i and j, resulting in a contradiction when combined with Equation (7.2). ∎

Before we derive our corollary, let us explicitly give the SA axioms of Bin-OP$_n$.

$$
\begin{gathered}
\textit{self}: \nu_{i,i} = 0 \quad \forall\, i \in n \\
\textit{trans}: \nu_{i,k} - \nu_{i,j} - \nu_{j,k} + 1 \geq 0 \quad \forall\, i,j,k \in [n] \\
\textit{impl}: \textstyle\sum_{i\in[\log n]}\omega_{i,j}^{1-a_i} + \nu_{j,a} \geq 0 \quad \forall\, j \in [n] \\
\text{where } a_1 \ldots a_{\log n} = \mathrm{bin}(a)
\end{gathered}
$$

Corollary 7.3. *The binary encoding of the Ordering Principle,* Bin-OP$_n$*, has* SA *rank at most* $2\log n$ *and* SA *size at most polynomial in* n*.*

Proof. Immediate from Lemma 7.1. ∎

8. SA+Squares. In this section we consider a proof system, SA+Squares, based on inequalities of multilinear polynomials. We now consider axioms as degree-1 polynomials in some set of variables and refutations as polynomials in those same variables. Then this system is gotten from SA by allowing addition of (linearised) squares of polynomials. In terms of strength this system will be strictly stronger than SA and at most as strong as Lasserre (also known as Sum-of-Squares), although we do not at this point see an exponential separation between SA+Squares and Lasserre. See [48, 49, 12] for more on the Lasserre proof system and [50] for tight degree lower bound results.

Consider the polynomial $w_{i,j}v_{i,j} - w_{i,j}v_{i,k}$. The square of this is

$$w_{i,j}v_{i,j}w_{i,j}v_{i,j} + w_{i,j}v_{i,k}w_{i,j}v_{i,k} - 2w_{i,j}v_{i,j}w_{i,j}v_{i,k}.$$

Using idempotence this linearises to $w_{i,j}v_{i,j} + w_{i,j}v_{i,k} - 2w_{i,j}v_{i,j}v_{i,k}$. Thus we know that this last polynomial is non-negative for all 0/1 settings of the variables.
A *degree-d* SA+Squares refutation of a set of linear inequalities (over terms) $q_1 \geq 0, \ldots, q_x \geq 0$ is an equation of the form

$$\sum_{i=1}^{x}p_iq_i + \sum_{i=1}^{y}r_i^2 = -1 \qquad (8.1)$$

where the p_i are polynomials with nonnegative coefficients and the degree of the polynomials $p_i q_i, r_i^2$ is at most d. We want to underline that we now consider a (product) term like $w_{i,j} v_{i,j} v_{i,k}$ as a product of its constituent variables, that is genuinely a term in the sense of part of a polynomial. This is opposed to the preceding sections in which we viewed it as a single variable $Z_{w_{i,j} \wedge v_{i,j} \wedge v_{i,k}}$. The translation from the degree discussed here to SA rank previously introduced may be paraphrased by "rank = degree − 1".

We note that the unary PHP_n^{n+1} becomes easy in this stronger proof system (see, e.g., Example 2.1 in [37]) while we shall see that the LOP_n remains hard (in terms of degree). The following is based on Example 2.1 in [37].

THEOREM 8.1. *The* Bin-PHP_n^{n+1} *has an* SA + Squares *refutation of degree* $2\log n + 1$ *and size* $O(n^3)$.

Proof. For short let $m = n + 1$ denote the number of pigeons. We begin by squaring the polynomial

$$1 - \sum_{i=1}^{m} \prod_{j=1}^{\log n} \omega_{i,j}^{a_j}$$

to get the degree $2\log n$, size quadratic in m inequality

$$1 - 2\sum_{i=1}^{m} \prod_{j=1}^{\log n} \omega_{i,j}^{a_j} + \sum_{1 \le i,i' \le m} \left(\prod_{j=1}^{\log n} \omega_{i,j}^{a_j}\right) \left(\prod_{j=1}^{\log n} \omega_{i',j}^{a_j}\right) \ge 0 \tag{8.2}$$

for every hole $a \in [n]$. On the other hand, by lifting each axiom

$$\sum_{j=1}^{\log n} \omega_{i,j}^{1-a_j} + \sum_{j=1}^{\log n} \omega_{i',j}^{1-a_j} \ge 1 \qquad \text{(whenever } i \ne i'\text{)}$$

by $\left(\prod_{j=1}^{\log n} \omega_{i,j}^{a_j}\right)\left(\prod_{j=1}^{\log n} \omega_{i',j}^{a_j}\right)$ we find $0 \ge \left(\prod_{j=1}^{\log n} \omega_{i,j}^{a_j}\right)\left(\prod_{j=1}^{\log n} \omega_{i',j}^{a_j}\right)$, in degree $2\log n + 1$. Adding these inequalities to (8.2) gives

$$1 - \sum_{i=1}^{m} \prod_{j=1}^{\log n} \omega_{i,j}^{a_j} \ge 0$$

in size again quadratic in m. Iterating this for every hole $a \in [n]$ we find

$$n - \sum_{a=1}^{n} \sum_{i=1}^{m} \prod_{j=1}^{\log n} \omega_{i,j}^{a_j} \ge 0 \tag{8.3}$$

in cubic size.

Note that for any pigeon $i \in [m]$, we can find in SA the linearly sized equality

$$\sum_{a=1}^{n} \prod_{j=1}^{\log n} \omega_{i,j}^{a_j} = 1. \tag{8.4}$$

in size linear in n.

This is done by induction on the number of bits involved (the range of j in the summation). For the base case of just $j = 1$ we clearly have

$$\omega_{i,1} + (1 - \omega_{i,1}) = 1.$$

Now suppose that for $k < \log n$, we have $\sum_{a\in[2^k]}\prod_{j=1}^{k}\omega_{i,j}^{a_j} = 1$. Multiplying both sides by $1 = \omega_{i,(k+1)} + (1-\omega_{i,(k+1)})$ gets the inductive step. The final term is of size $O(2^{\log n}) = O(n)$.

Summing 8.4 for every such hole i we find

$$\sum_{i=1}^{m}\sum_{a=1}^{n}\prod_{j=1}^{\log n}\omega_{i,j}^{a_j} \geq m. \tag{8.5}$$

Adding 8.5 to 8.3, we get the desired contradiction, $n - m \geq 0$. ∎

This last theorem, combined with the exponential SA size lower bound given in Theorem 6.5, shows us that SA+Squares is exponentially separated from SA in terms of size.

We now turn our attention to LOP_n, whose SA axioms we reproduce to refresh the reader's memory.

$$\begin{gathered}
\mathit{self} : v_{i,i} = 0 \quad \forall\, i \in n \\
\mathit{trans} : v_{i,k} - v_{i,j} - v_{j,k} + 1 \geq 0 \quad \forall\, i,j,k \in [n] \\
\mathit{impl} : v_{i,j} - w_{i,j} \geq 0 \quad \forall\, i,j \in [n] \\
\mathit{total} : v_{i,j} + v_{j,i} - 1 \geq 0 \quad \forall\, i \neq j \in [n] \\
\mathit{lower} : \textstyle\sum_{i\in[n]} w_{i,j} - 1 \geq 0 \quad \forall\, j \in [n]
\end{gathered}$$

We give our lower bound for the unary LOP_n by producing a linear function val (which we will call a *valuation*) from terms into $\mathbb{R}$ such that

1. for each axiom $p \geq 0$ and every term X with $deg(Xp) \leq d$ we have $\mathsf{val}(Xp) \geq 0$, and
2. we have $\mathsf{val}(r^2) \geq 0$ whenever $deg(r^2) \leq d$.
3. $\mathsf{val}(1) = 1$.

The existence of such a valuation clearly implies that a degree-d SA+Squares refutation cannot exist, as it would result in a contradiction when applied to both sides of (8.1).

To verify that $\mathsf{val}(r^2) \geq 0$ whenever $deg(r^2) \leq d$ we show that the so-called *moment-matrix* $\mathcal{M}_{\mathsf{val}}$ is positive semidefinite. The degree-d moment matrix is defined to be the symmetric square matrix whose rows and columns are indexed by terms of size at most $d/2$ and each entry is the valuation of the product of the two terms indexing that entry. Given any polynomial σ of degree at most $d/2$ let c be its vector of coefficients. Then if $\mathcal{M}_v$ is positive semidefinite:

$$\mathsf{val}(\sigma^2) = \sum_{deg(T_1),deg(T_2)\leq d/2} c(T_1)c(T_2)v(T_1T_2) = c^\top \mathcal{M}_v c \geq 0.$$

(For more on this see e.g. [48], section 2.)

Theorem 8.2. *There is no* SA + Squares *refutation of the (unary)* LOP_n *with degree at most* $(n-3)/2$.

Proof. For each term T, let $\mathsf{val}(T)$ be the probability that T is consistent with a permutation on the n elements taken uniformly at random or, in other words, the number of permutations consistent with T divided by $n!$. Here we view $w_{x,y}$ as equal to $v_{x,y}$. This valuation trivially satisfies the lifts of the *self*, *trans* and *total* axioms as they are satisfied by each permutation (linear order). It satisfies the lifts of the *impl* axioms by construction. We now claim that the lifts of the *lower* axioms (those containing only w variables) of degree up to $\frac{n-3}{2}$ are also satisfied by $v\,(.)$. Indeed, let us consider the lifting by T of the *lower* axiom for x

$$\sum_{y=1}^{n} Tw_{x,y} \geq T. \tag{8.6}$$

Since T mentions at most $n-3$ elements, there must be at least two $y_1 \neq y_2$ that are different from all of them and from x. For any permutation that is consistent with T, the probability that each of the y_1 and y_2 is smaller than x is precisely a half, and thus

$$\mathsf{val}\,(Tw_{x,y_1}) + \mathsf{val}\,(Tw_{x,y_2}) = \mathsf{val}\,(T)\,.$$

Therefore the valuation of the LHS of (8.6) is always greater than or equal to the valuation of T.

Finally, we need to show that the valuation is consistent with the non-negativity of (the linearisation of) any squared polynomial. It is easy to see that the moment matrix for val can be written as

$$\frac{1}{n!}\sum_{\sigma} V_\sigma V_\sigma^T$$

where the summation is over all permutations on n elements and for a permutation σ, V_σ is its characteristic vector. The characteristic vector of a permutation σ is a Boolean column vector indexed by terms and whose entries are 1 or 0 depending on whether the respective index term is consistent or not with the permutation σ. Clearly the moment matrix is positive semidefinite being a sum of (rank one) positive semidefinite matrices. ∎

The previous theorem is interesting because a degree upper bound in Lasserre of order $\sqrt{n}\log n$ is known for LOP_n [57]. It is proved for a slightly different formulation of LOP_n from ours, but it is readily seen to be equivalent to our formulation and we provide the translation in the appendix. Thus, Theorem 8.2, together with [57], shows a quadratic rank separation between SA+Squares and Lasserre.

9. Contrasting unary and binary encodings. To work with a more general theory in which to contrast the complexity of refuting the binary and unary versions of combinatorial principles, following Riis [64] we consider principles which are expressible as first order formulas with no finite model in Π_2-form, i.e. as $\forall\vec{x}\exists\vec{w}\varphi(\vec{x},\vec{w})$ where $\varphi(\vec{x},\vec{y})$ is a formula built on a family of relations $\vec{R}$. For example, we already met the Ordering Principle, that states that a finite partial order has a maximal element. Its negation can be expressed in Π_2-form as in Section 7 as: $\forall x,y,z\exists w\ \neg R(x,x)\wedge(R(x,y)\wedge R(y,z)\rightarrow R(x,z))\wedge R(x,w)$. This can be translated into a unsatisfiable CNF using a *unary encoding* of the witness, as already discussed in Section 7.

As a second example we consider the Pigeonhole Principle which states that a total mapping from $[m]$ to $[n]$ has necessarily a collision when m and n are integers with $m>n$. Following Riis [64], for $m=n+1$, the negation of its relational form can be expressed as a Π_2-formula as

$$\forall x,y,z\exists w\ \neg R(x,0)\wedge(R(x,z)\wedge R(y,z)\rightarrow x=y)\wedge R(x,w)$$

and its usual unary and binary propositional encoding have already been introduced. Notice that in the case of Pigeonhole Principle, the existential witness w to the type *pigeon* is of the distinct type *hole*. Furthermore, pigeons only appear on the left-hand side of atoms $R(x, z)$ and holes only appear on the right-hand side. For the Ordering Principle instead, the transitivity axioms effectively enforce the type of y appears on both the left- and right-hand side of atoms $R(x, z)$. This accounts for why, in the case of the Pigeonhole Principle, we did not need to introduce any new variables to give the binary encoding, yet for the Ordering Principle a new variable w appears.

9.1. Binary encodings of principles versus their unary functional encodings. Recall the unary functional encoding of a combinatorial principle C, denoted Un-Fun-C(n), replaces the big clauses from Un-C(n), of the form $v_{i,1} \vee \ldots \vee v_{i,n}$, with $v_{i,1} + \ldots + v_{i,n} = 1$, where addition is made on the natural numbers. This is equivalent to augmenting the axioms $\neg v_{i,j} \vee \neg v_{i,k}$, for $j \neq k \in [n]$.

LEMMA 9.1. *Suppose there is a Resolution refutation of* Bin-C(n) *of size* $S(n)$*. Then there is a Resolution refutation of* Un-Fun-C(n) *of size at most* $n^2 \cdot S(n)$*.*

Proof. Take a decision DAG π' for Bin-C(n), where, without loss of generality, n is even, and consider the point at which some variable $\nu_{i,j}$ is questioned. Each node in π' will be expanded to a small tree in π, which will be a decision DAG for Un-Fun-C(n). The question "$\nu_{i,j}$?" in π will become a sequence of questions $v_{i,1}, \ldots, v_{i,n}$ where we stop the small tree when one of these is answered true, which must eventually happen because if they are all answered false we contradict an axiom. Suppose $v_{i,k}$ is true. If the jth bit of k is 1 we ask now all $v_{i,b_1}, \ldots, v_{i,b_{\frac{n}{2}}}$, where $b_1, \ldots, b_{\frac{n}{2}}$ are precisely the numbers in $[n]$ whose jth bit is 0. All of these must be false. Likewise, if the jth bit of k is 0 we ask all $v_{i,b_1}, \ldots, v_{i,b_{\frac{n}{2}}}$, where $b_1, \ldots, b_{\frac{n}{2}}$ are precisely the numbers whose jth bit is 1. All of these must be false. We now unify the branches on these two possibilities, forgetting any intermediate information. (To give an example, suppose $j = 2$. Then the two outcomes are $\neg v_{i,1} \wedge \neg v_{i,3} \wedge \ldots \wedge \neg v_{i,n-1}$ and $\neg v_{i,2} \wedge \neg v_{i,4} \wedge \ldots \wedge \neg v_{i,n}$.) Thus, π' gives rise to π of size $n^2 \cdot S(n)$ and the result follows. ∎

9.2. The Ordering Principle in binary. Recall the Ordering Principle whose binary formulation Bin-OP$_n$ we met in Section 7.

LEMMA 9.2. Bin-OP$_n$ *has refutations in Resolution of size* $O(n^3)$*.*

Proof. We follow the well-known proof for the unary version of the Ordering Principle, from [67]. Consider the domain to be $[n] = \{1, \ldots, n\}$. At the ith stage of the decision DAG we will find a maximal element, ordered by R, among $[i] = \{1, \ldots, i\}$. That is, we will have a CNF record of the *special* form

$$\neg \nu_{j,1} \wedge \ldots \wedge \neg \nu_{j,j-1} \wedge \neg \nu_{j,j+1} \wedge \ldots \wedge \neg \nu_{j,i}$$

for some $j \in [i]$. The base case $i = 1$ is trivial. Let us explain the inductive step. From the displayed CNF record above we ask the question $\nu_{j,i+1}$? If $\nu_{j,i+1}$ is true, then ask the sequence of questions $\nu_{i+1,1}, \ldots, \nu_{i+1,i}$, all of which must be false by transitivity (the case $i = j$ uses irreflexivity too). Now, by forgetting information, we uncover a new CNF record of the special form. Suppose now $\nu_{j,i+1}$ is false. Then we equally have a new CNF record again in the special form. Let us consider the size of our decision tree so far. There are n^2 nodes corresponding to special CNF records and navigating between special CNF records involves a path of length n, so we have a DAG of size n^3. Finally, at $i = n$, we have a CNF record of the form

$$\neg \nu_{j,1} \wedge \ldots \wedge \neg \nu_{j,j-1} \wedge \neg \nu_{j,j+1} \wedge \ldots \wedge \neg \nu_{j,n}.$$

Now we expand a tree questioning the sequence $\omega_{j,1}, \ldots, \omega_{j,\log n}$, and discover each leaf labels a contradiction of the clauses of the final type. We have now added $n \cdot 2^{\log n}$ nodes, so our final DAG is of size at most $n^3 + n^2$. $\square$

10. Final remarks. In this paper we started a systematic study of binary encodings of combinatorial principles in proof complexity. Various questions arise directly from our exposition. Primarily, there is the question as to the optimality of our lower bounds for the binary encodings of k-Clique and the (weak) Pigeonhole Principle. In terms of the strongest refutation system Res(s) (largest s) for which we can prove superpolynomial bounds, then it is not hard to see that our method can go no further than $s = o((\log \log n)^{\frac{1}{3}})$ for the former, and $s = O(\log^{\frac{1}{2}-\epsilon} n)$ for the latter. This is because we run out of space with the random restrictions as they become nested in the induction. We have no reason, however to think that our results are truly optimal, only that another method is needed to improve them.

A second question about binary encodings concern width and rank. From our work it holds that in SA the unary encoding can be harder than binary with respect to rank. One might question whether the same hold for Resolution width. Are there formulas that require large width in the unary encoding, but can refuted in small width in the binary encoding? Notice that in the other direction a large separation is not possible. In particular it is straightforward to see that if the unary version of a formula F over n variables has Resolution refutations of size S and width w, then the binary version of F has Resolution refutations of size $Sw^{\log n}$ and width $w \log n$.

Other questions concern to what extent the converses of our lemmas might hold. The converse of Lemma 9.1 (even for n^2 replaced by some sublinear polynomial) is false. For example, consider the very weak Pigeonhole Principle of [24]. However, this example is somewhat disingenuous as the parameter n is no longer polynomially related to the number of pigeons m and the size of the clause set.

Finally an important question, not strictly regarding binary encodings, is the relative efficiency of SA+Squares with respect to Lasserre. Is there a meaningful size separation between SA+Squares and Lasserre? Is Lasserre strictly stronger? At present we know only the quadratic rank separation implied by our $\Omega(n)$ (Theorem 8.2) lower bound in SA + Squares and Potechin's upper $O(\sqrt{n})$ upper bound in Lasserre for LOP_n .

Acknowledgments. We are grateful to Ilario Bonacina for reading a preliminary version of this work and addressing us some useful comments and observations. We are further grateful to several anonymous reviewers for detailed corrections and comments. We are grateful in particular to two reviewers in the journal version who improved greatly our exposition.

REFERENCES

[1] M. Alekhnovich, *Lower bounds for k-DNF resolution on random 3-CNFs*, Computational Complexity, 20 (2011), pp. 597–614, https://doi.org/10.1007/s00037-011-0026-0, https://doi.org/10.1007/s00037-011-0026-0.

[2] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov, and A. Wigderson, *Space complexity in propositional calculus*, SIAM J. Comput., 31 (2002), pp. 1184–1211, https://doi.org/10.1137/S0097539700366735, https://doi.org/10.1137/S0097539700366735.

[3] A. Atserias, *Improved bounds on the weak pigeonhole principle and infinitely many primes from weaker axioms*, Theor. Comput. Sci., 295 (2003), pp. 27–39, https://doi.org/10.1016/S0304-3975(02)00394-8, https://doi.org/10.1016/S0304-3975(02)00394-8.

[4] A. Atserias, I. Bonacina, S. F. de Rezende, M. Lauria, J. Nordström, and A. A. Razborov, *Clique is hard on average for regular resolution*, in Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018, I. Diakonikolas, D. Kempe, and M. Henzinger, eds., ACM, 2018, pp. 866–877, https://doi.org/10.1145/3188745.3188856, http://doi.acm.org/10.1145/3188745.3188856.

[5] A. Atserias, I. Bonacina, S. F. de Rezende, M. Lauria, J. Nordström, and A. A. Razborov, *Clique is hard on average for regular resolution*, CoRR, abs/2012.09476 (2020), https://arxiv.org/abs/2012.09476, https://arxiv.org/abs/2012.09476.

[6] A. Atserias, M. L. Bonet, and J. L. Esteban, *Lower bounds for the weak pigeonhole principle and random formulas beyond resolution*, Inf. Comput., 176 (2002), pp. 136–152, https://doi.org/10.1006/inco.2002.3114, https://doi.org/10.1006/inco.2002.3114.

[7] A. Atserias and V. Dalmau, *A combinatorial characterization of resolution width*, J. Comput. Syst. Sci., 74 (2008), pp. 323–334, https://doi.org/10.1016/j.jcss.2007.06.025, https://doi.org/10.1016/j.jcss.2007.06.025.

[8] A. Atserias, M. Lauria, and J. Nordström, *Narrow proofs may be maximally long*, CoRR, abs/1409.2731 (2014), http://arxiv.org/abs/1409.2731, https://arxiv.org/abs/1409.2731.

[9] A. Atserias, M. Lauria, and J. Nordström, *Narrow proofs may be maximally long*, ACM Trans. Comput. Log., 17 (2016), pp. 19:1–19:30, https://doi.org/10.1145/2898435, https://doi.org/10.1145/2898435.

[10] A. Atserias and M. Müller, *Automating resolution is NP-hard*, Journal of the ACM, 67 (2020), pp. 31:1–31:17, https://doi.org/10.1145/3409472, https://doi.org/10.1145/3409472.

[11] A. Atserias, M. Müller, and S. Oliva, *Lower bounds for DNF-refutations of a relativized weak pigeonhole principle*, J. Symb. Log., 80 (2015), pp. 450–476, https://doi.org/10.1017/jsl.2014.56, https://doi.org/10.1017/jsl.2014.56.

[12] B. Barak and D. Steurer, *Sum-of-squares proofs and the quest toward optimal algorithms*, in Proceedings of International Congress of Mathematicians (ICM), vol. IV, 2014, pp. 509–533.

[13] P. Beame, R. Impagliazzo, and A. Sabharwal, *Resolution complexity of independent sets in random graphs*, in Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, Illinois, USA, June 18-21, 2001, IEEE Computer Society, 2001, pp. 52–68, https://doi.org/10.1109/CCC.2001.933872, https://doi.org/10.1109/CCC.2001.933872.

[14] P. Beame and T. Pitassi, *Simplified and improved resolution lower bounds*, in 37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996, IEEE Computer Society, 1996, pp. 274–282, https://doi.org/10.1109/SFCS.1996.548486, https://doi.org/10.1109/SFCS.1996.548486.

[15] E. Ben-sasson and A. Wigderson, *Short proofs are narrow - resolution made simple*, in Journal of the ACM, 1999, pp. 517–526.

[16] O. Beyersdorff, N. Galesi, and M. Lauria, *A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover-delayer games*, Inf. Process. Lett., 110 (2010), pp. 1074–1077, https://doi.org/10.1016/j.ipl.2010.09.007, http://dx.doi.org/10.1016/j.ipl.2010.09.007.

[17] O. Beyersdorff, N. Galesi, and M. Lauria, *Parameterized complexity of DPLL search procedures*, ACM Trans. Comput. Logic, 14 (2013), pp. 20:1–20:21, https://doi.org/10.1145/2499937.2499941, http://doi.acm.org/10.1145/2499937.2499941.

[18] O. Beyersdorff, N. Galesi, M. Lauria, and A. A. Razborov, *Parameterized bounded-depth frege is not optimal*, TOCT, 4 (2012), pp. 7:1–7:16, https://doi.org/10.1145/2355580.2355582, http://doi.acm.org/10.1145/2355580.2355582.

[19] B. Bollobás, *Threshold functions for small subgraphs*, Math. Proc. Cambridge Philos. Soc., 90 (1980), pp. 197–206.

[20] B. Bollobás, *Random Graphs*, Cambridge University Press, 2001, https://doi.org/10.1017/cbo9780511814068, https://doi.org/10.1017%2Fcbo9780511814068.

[21] I. Bonacina and N. Galesi, *A framework for space complexity in algebraic proof systems*, J. ACM, 62 (2015), pp. 23:1–23:20, https://doi.org/10.1145/2699438, http://doi.acm.org/10.1145/2699438.

[22] I. Bonacina, N. Galesi, and N. Thapen, *Total space in resolution*, SIAM J. Comput., 45 (2016), pp. 1894–1909, https://doi.org/10.1137/15M1023269, https://doi.org/10.1137/15M1023269.

[23] M. L. Bonet and N. Galesi, *Optimality of size-width tradeoffs for resolution*, Computational Complexity, 10 (2001), pp. 261–276, https://doi.org/10.1007/s000370100000, https://doi.

org/10.1007/s000370100000.

[24] S. R. Buss and T. Pitassi, *Resolution and the weak pigeonhole principle*, in Computer Science Logic, 11th International Workshop, CSL '97, Annual Conference of the EACSL, Aarhus, Denmark, August 23-29, 1997, Selected Papers, 1997, pp. 149–156, https://doi.org/10.1007/BFb0028012, http://dx.doi.org/10.1007/BFb0028012.

[25] V. Chvátal, *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Math., 4 (1973), pp. 305–337.

[26] S. S. Dantchev, *Rank complexity gap for Lovász-Schrijver and Sherali-Adams proof systems*, in STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, New York, NY, USA, 2007, ACM Press, pp. 311–317, https://doi.org/http://doi.acm.org/10.1145/1250790.1250837.

[27] S. S. Dantchev, N. Galesi, and B. Martin, *Resolution and the binary encoding of combinatorial principles*, in 34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA., 2019, pp. 6:1–6:25, https://doi.org/10.4230/LIPIcs.CCC.2019.6, https://doi.org/10.4230/LIPIcs.CCC.2019.6. See http://arxiv.org/abs/1809.02843.

[28] S. S. Dantchev, A. Ghani, and B. Martin, *Sherali-adams and the binary encoding of combinatorial principles*, in LATIN 2020: Theoretical Informatics - 14th Latin American Symposium, São Paulo, Brazil, January 5-8, 2021, Proceedings, Y. Kohayakawa and F. K. Miyazawa, eds., vol. 12118 of Lecture Notes in Computer Science, Springer, 2020, pp. 336–347, https://doi.org/10.1007/978-3-030-61792-9_27, https://doi.org/10.1007/978-3-030-61792-9_27.

[29] S. S. Dantchev, B. Martin, and M. N. C. Rhodes, *Tight rank lower bounds for the Sherali-Adams proof system*, Theor. Comput. Sci., 410 (2009), pp. 2054–2063, https://doi.org/10.1016/j.tcs.2009.01.002, https://doi.org/10.1016/j.tcs.2009.01.002.

[30] S. S. Dantchev and S. Riis, *Tree resolution proofs of the weak pigeon-hole principle*, in Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, Illinois, USA, June 18-21, 2001, 2001, pp. 69–75, https://doi.org/10.1109/CCC.2001.933873, http://doi.ieeecomputersociety.org/10.1109/CCC.2001.933873.

[31] S. S. Dantchev and S. Riis, *On relativisation and complexity gap*, in Computer Science Logic, 17th International Workshop, CSL 2003, 12th Annual Conference of the EACSL, and 8th Kurt Gödel Colloquium, KGC 2003, Vienna, Austria, August 25-30, 2003, Proceedings, M. Baaz and J. A. Makowsky, eds., vol. 2803 of Lecture Notes in Computer Science, Springer, 2003, pp. 142–154, https://doi.org/10.1007/978-3-540-45220-1_14, https://doi.org/10.1007/978-3-540-45220-1_14.

[32] S. F. de Rezende, M. Göös, J. Nordström, T. Pitassi, R. Robere, and D. Sokolov, *Automating algebraic proof systems is NP-hard*, Electron. Colloquium Comput. Complex. (To appear in STOC 2021), 27 (2020), p. 64, https://eccc.weizmann.ac.il/report/2020/064.

[33] J. L. Esteban, N. Galesi, and J. Messner, *On the complexity of resolution with bounded conjunctions*, Theor. Comput. Sci., 321 (2004), pp. 347–370, https://doi.org/10.1016/j.tcs.2004.04.004, https://doi.org/10.1016/j.tcs.2004.04.004.

[34] Y. Filmus, M. Lauria, J. Nordström, N. Ron-Zewi, and N. Thapen, *Space complexity in polynomial calculus*, SIAM J. Comput., 44 (2015), pp. 1119–1153, https://doi.org/10.1137/120895950, https://doi.org/10.1137/120895950.

[35] N. Galesi and M. Lauria, *Optimality of size-degree tradeoffs for polynomial calculus*, ACM Trans. Comput. Log., 12 (2010), pp. 4:1–4:22, https://doi.org/10.1145/1838552.1838556, https://doi.org/10.1145/1838552.1838556.

[36] R. E. Gomory, *Solving linear programming problems in integers*, in Combinatorial Analysis, Proceedings of Symposia in Applied Mathematics, R. Bellman and M. Hall, eds., vol. 10, Providence, RI, 1960.

[37] D. Grigoriev, E. A. Hirsch, and D. V. Pasechnik, *Complexity of semi-algebraic proofs*, in STACS '02: Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science, London, UK, 2002, Springer-Verlag, pp. 419–430.

[38] A. Haken, *The intractability of resolution*, Theor. Comput. Sci., 39 (1985), pp. 297–308.

[39] P. Hall, *On Representatives of Subsets*, Journal of the London Mathematical Society, s1-10 (1935), pp. 26–30, https://doi.org/10.1112/jlms/s1-10.37.26, https://doi.org/10.1112/jlms/s1-10.37.26, https://arxiv.org/abs/https://academic.oup.com/jlms/article-pdf/s1-10/1/26/6471457/s1-10-37-26.pdf.

[40] J. Håstad, *Computational Limitations for Small Depth Circuits*, MIT Press, 1987.

[41] P. Hrubeš and P. Pudlák, *Random formulas, monotone circuits, and interpolation*, in 58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017, C. Umans, ed., IEEE Computer Society, 2017, pp. 121–131, https://doi.org/10.1109/FOCS.2017.20, https://doi.org/10.1109/FOCS.2017.20.

[42] D. Itsykson and A. Riazanov, *Proof complexity of natural formulas via communication arguments*, Electron. Colloquium Comput. Complex., 27 (2020), p. 184, https://eccc.weizmann.ac.il/report/2020/184.
[43] J. Krajíček, *Bounded arithmetic, propositional logic and complexity theory*, Cambridge University Press, 1995.
[44] J. Krajíček, *On the weak pigeonhole principle*, Fundamenta Mathematicae, 170 (2001), pp. 123–140.
[45] B. Krishnamurthy, *Short proofs for tricky formulas*, Acta Inf., 22 (1985), pp. 253–275, https://doi.org/10.1007/BF00265682, https://doi.org/10.1007/BF00265682.
[46] O. Kullmann, *Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs*, Electronic Colloquium on Computational Complexity (ECCC), (1999), http://eccc.hpi-web.de/eccc-reports/1999/TR99-041/index.html.
[47] G. Kwon and W. Klieber, *Efficient CNF encoding for selecting 1 from n objects*, in Fourth Workshop on Constraints in Formal Verification (CFV '07), 2007.
[48] J. B. Lasserre, *An explicit exact SDP relaxation for nonlinear 0-1 programs*, in Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO01), K. Aardal and B. Gerards, eds., vol. 2081 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2001, pp. 293–303.
[49] M. Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for* $0-1$ *programming*, Tech. Report PNA–R0108, Amsterdam, 2001.
[50] M. Lauria and J. Nordström, *Tight size-degree bounds for sums-of-squares proofs*, computational complexity, 26 (2017), pp. 911–948, https://doi.org/10.1007/s00037-017-0152-4, https://doi.org/10.1007/s00037-017-0152-4.
[51] M. Lauria, P. Pudlák, V. Rödl, and N. Thapen, *The complexity of proving that a graph is ramsey*, Combinatorica, 37 (2017), pp. 253–268, https://doi.org/10.1007/s00493-015-3193-9, https://doi.org/10.1007/s00493-015-3193-9.
[52] M. Lauria, P. Pudlák, V. Rödl, and N. Thapen, *The complexity of proving that a graph is ramsey*, Combinatorica, 37 (2017), pp. 253–268, https://doi.org/10.1007/s00493-015-3193-9, https://doi.org/10.1007/s00493-015-3193-9.
[53] L. Lovász and A. Schrijver, *Cones of matrices and set-functions and* 0-1 *optimization*, SIAM J. Optimization, 1 (1991), pp. 166–190.
[54] A. Maciel, T. Pitassi, and A. R. Woods, *A new proof of the weak pigeonhole principle*, J. Comput. Syst. Sci., 64 (2002), pp. 843–872, https://doi.org/10.1006/jcss.2002.1830, https://doi.org/10.1006/jcss.2002.1830.
[55] M. Mitzenmacher and E. Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press, 2005.
[56] J. Petke, *Bridging Constraint Satisfaction and Boolean Satisfiability*, Artificial Intelligence: Foundations, Theory, and Algorithms, Springer, 2015, https://doi.org/10.1007/978-3-319-21810-6, http://dx.doi.org/10.1007/978-3-319-21810-6.
[57] A. Potechin, *Sum of squares bounds for the ordering principle*, in Proceedings of the 35th Computational Complexity Conference, 2020, pp. 1–37.
[58] P. Pudlák, *Proofs as games*, American Mathematical Monthly, (2000), pp. 541–550.
[59] R. Raz, *Resolution lower bounds for the weak pigeonhole principle*, J. ACM, 51 (2004), pp. 115–138, https://doi.org/10.1145/972639.972640, http://doi.acm.org/10.1145/972639.972640.
[60] A. Razborov, *Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution*, Annals of Mathematics, 181 (2015), pp. 415–472.
[61] A. A. Razborov, *Proof complexity of pigeonhole principles*, in Developments in Language Theory, W. Kuich, G. Rozenberg, and A. Salomaa, eds., Berlin, Heidelberg, 2002, Springer Berlin Heidelberg, pp. 100–116.
[62] A. A. Razborov, *Resolution lower bounds for the weak functional pigeonhole principle*, Theor. Comput. Sci., 1 (2003), pp. 233–243, https://doi.org/10.1016/S0304-3975(02)00453-X, https://doi.org/10.1016/S0304-3975(02)00453-X.
[63] M. Rhodes, *Rank lower bounds for the Sherali-Adams operator*, in CiE, S. B. Cooper, B. Löwe, and A. Sorbi, eds., vol. 4497 of Lecture Notes in Computer Science, Springer, 2007, pp. 648–659.
[64] S. Riis, *A complexity gap for tree resolution*, Computational Complexity, 10 (2001), pp. 179–209.
[65] N. Segerlind, S. R. Buss, and R. Impagliazzo, *A switching lemma for small restrictions and lower bounds for k-DNF resolution*, SIAM J. Comput., 33 (2004), pp. 1171–1200, https://doi.org/10.1137/S0097539703428555, https://doi.org/10.1137/S0097539703428555.
[66] H. D. Sherali and W. P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM J. Discrete Math.,

3 (1990), pp. 411–430.

[67] G. Stålmarck, *Short resolution proofs for a sequence of tricky formulas*, Acta Inf., 33 (1996), pp. 277–280, https://doi.org/10.1007/s002360050044, https://doi.org/10.1007/s002360050044.

[68] N. Thapen and A. Skelley, *The provably total search problems of bounded arithmetic*, Proceedings of the London Mathematical Society, 103 (2011), pp. 106–138.

[69] J. H. van Lint and R. M. Wilson, *A course in combinatorics*, 1992.

[70] T. Walsh, *SAT v CSP*, in Principles and Practice of Constraint Programming - CP 2000, 6th International Conference, Singapore, September 18-21, 2000, Proceedings, 2000, pp. 441–456, https://doi.org/10.1007/3-540-45349-0_32, https://doi.org/10.1007/3-540-45349-0_32.

11. Appendix.

11.1. Potechin's encoding of LOP_n**.** Potechin provides a $O(\sqrt{n}\log n)$ upper bound in Lasserre for the following formulation of the linear ordering principle, which we purposefully give in the variables $x_{i,j}$ instead of our $v_{i,j}$.

$$x_{i,j} + x_{j,i} = 1 \qquad \text{for all distinct } i,j \in [n]$$
$$x_{i,j}x_{j,k}(1 - x_{i,k}) = 0 \qquad \text{for all distinct } i,j,k \in [n]$$
$$\sum_{i\in[n],i\neq i} x_{i,j} = 1 + z_j^2$$

Note that anything we can prove using transitivity of the form $x_{i,j}x_{j,k}(1 - x_{i,k}) = 0$ we can prove using $v_{i,k} - v_{i,j} - v_{j,k} \geq -1$. That $v_{i,j}v_{j,k} \geq v_{i,j}v_{j,k}v_{i,k}$ comes from monotonicity, and the opposite inequality comes from lifting by $v_{i,j}v_{j,k}$:

$$-v_{i,j}v_{j,k} \leq v_{i,j}v_{j,k}v_{i,k} - 2v_{i,j}v_{j,k} \implies v_{i,j}v_{j,k} \leq v_{i,j}v_{j,k}v_{i,k}.$$

Potechin's proof moves along the following lines. Define an operator E on terms that behaves the same as the val used in Theorem 8.2, but

1. If some z_j appears with degree 1 in T, then $E[T] = 0$, and
2. If T is of the form $z_j^2 T'$ for some j and T', $E[T] = E\left[\left(\sum_{i\in[n],i\neq i} x_{ij} - 1\right)T'\right]$

Potechin proves the following.

Lemma 11.1 (Lemma 4.2 in [57]). *There exists a polynomial g, only in the variables $x_{i,j}$ and of degree $O(\sqrt{n}\log n)$ such that*

$$E\left[\left(\sum_{i\neq j} x_{i,j} - 1\right)g^2\right] = \mathit{val}\left(\left(\sum_{i\neq j} x_{i,j} - 1\right)g^2\right) < 0.$$

Potechin then proves the following Lasserre identity using only the totality and transitivity axioms (which exist also in our formulation). Note S_k is the symmetric group on the elements of $[k]$.

Lemma 11.2 (Lemma 4.7 in [57]). *For all $A = \{i_1, i_2, \ldots, i_k\} \subseteq [n]$, there exists a degree $k+2$ proof that*

$$\sum_{\pi\in S_k}\prod_{j=1}^{k-1} x_{i_{\pi(j)}i_{\pi(j+1)}} = 1.$$

Finally, Potechin proves that the 'symmetric group average' of a polynomial can be shown to be equal to its valuation.

LEMMA 11.3 (Lemma 4.8 in [57]). *For any polynomial p of degree d in the variables x_{ij}, there exists a proof of at most degree $3d+2$ that*

$$\frac{1}{n!}\sum_{\pi \in S_n} \pi(p) = \mathsf{val}(p)$$

(where the action of S_n is to permute the indices in the monomials of p).

Lemma 11.1 and 11.3 together furnish a Lasserre refutation of the required form.

11.2. Recapitulation of the unary and binary encodings of the main principles. This section is devoted solely to Figure 1.

principle	unary case	binary case
(Bin-)$\mathrm{Clique}^{\mathrm{k}}_{\mathrm{n}}$	$\neg v_{i,a} \vee \neg v_{j,b}$ whenever $\neg E((i,a),(j,b))$ and $\bigvee_{a\in[n]} v_{i,a}$ for each block $i \in [k]$	$(\omega_{i,1}^{1-a_1} \vee \ldots \vee \omega_{i,\log n}^{1-a_{\log n}})$ $\vee$ $(\omega_{j,1}^{1-b_1} \vee \ldots \vee \omega_{j,\log n}^{1-b_{\log n}})$ whenever $\neg E((i,a),(j,b))$ where binary representations are $a = a_1 \ldots a_{\log n}$ $b = b_1 \ldots b_{\log n}$
(Bin-)PHP^m_n	$\neg v_{i,a} \vee \neg v_{j,a}$ whenever $i \neq j$ and $\bigvee_{a\in[n]} v_{i,a}$ for each pigeon $i \in [m]$	$(\omega_{i,1}^{1-a_1} \vee \ldots \vee \omega_{i,\log n}^{1-a_{\log n}})$ $\vee$ $(\omega_{j,1}^{1-a_1} \vee \ldots \vee \omega_{j,\log n}^{1-a_{\log n}})$ whenever $i \neq j$ where binary representation is $a = a_1 \ldots a_{\log n}$
(Bin-)OPn	$\neg v_{i,i}$ for all $i \in [n]$ $\neg v_{i,j} \vee \neg v_{j,k} \vee v_{i,k}$ for all $i,j,k \in [n]$ and $\bigvee_{a\in[n]} v_{i,a}$ for all $a \in [n]$	$\neg \nu_{i,i}$ for all $i \in [n]$ $\neg \nu_{i,j} \vee \neg \nu_{j,k} \vee \nu_{i,k}$ for all $i,j,k \in [n]$ $\bigvee_{i\in[n]} \nu_{i,j}$ for all $j \in [n]$ and $(\omega_{i,1}^{1-a_1} \vee \ldots \vee \omega_{i,\log n}^{1-a_{\log n}} \vee \nu_{a,i})$ for all $a \in [n]$ whose binary representation is $a_1 \ldots a_{\log n}$

FIG. 1. *Recapitulation of the unary and binary encodings of the main principles.*