# Resolution and the Binary Encoding of Combinatorial Principles

**Stefan Dantchev**
Department of Computer Science University of Durham, UK
s.s.dantchev@durham.ac.uk

**Nicola Galesi**
Dipartimento di Informatica, Sapienza Università di Roma, Italy
nicola.galesi@uniroma1.it

**Barnaby Martin**
Department of Computer Science University of Durham, UK
barnaby.d.martin@durham.ac.uk

## Abstract

$\mathsf{Res}(s)$ is an extension of Resolution working on $s$-DNFs. We prove tight $n^{\Omega(k)}$ lower bounds for the size of refutations of the binary version of the $k$-Clique Principle in $\mathsf{Res}(o(\log\log n))$. Our result improves that of Lauria, Pudlák et al. [27] who proved the lower bound for $\mathsf{Res}(1)$, i.e. Resolution. The exact complexity of the (unary) $k$-Clique Principle in Resolution is unknown. To prove the lower bound we do not use any form of the Switching Lemma [35], instead we apply a recursive argument specific for binary encodings. Since for the $k$-Clique and other principles lower bounds in Resolution for the unary version follow from lower bounds in $\mathsf{Res}(\log n)$ for their binary version 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 $\mathsf{Bin\text{-}PHP}_n^m$ for $m > n$. Using the the same recursive approach we prove the new result that for any $\delta > 0$, $\mathsf{Bin\text{-}PHP}_n^m$ requires proofs of size $2^{n^{1-\delta}}$ in $\mathsf{Res}(s)$ for $s = o(\log^{1/2} n)$. Our lower bound is almost optimal since for $m \geq 2^{\sqrt{n \log n}}$ there are quasipolynomial size proofs of $\mathsf{Bin\text{-}PHP}_n^m$ in $\mathsf{Res}(\log n)$.

Finally we propose a general theory in which to compare the complexity of refuting the binary and unary versions of large classes of combinatorial principles, namely those expressible as first order formulae in $\Pi_2$-form and with no finite model.

**2012 ACM Subject Classification** Theory of computation → Complexity theory and logic

**Keywords and phrases** Proof complexity, $k$-DNF resolution, binary encodings, Clique and Pigeonhole principle

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2019.6

**Acknowledgements** 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 further corrections and comments.

## 1 Introduction

Various fundamental combinatorial principles used in Proof Complexity may be given in first-order logic as sentences $\varphi$ with no finite models. Riis discusses in [34] how to generate from $\varphi$ a family of CNFs, the $n$th of which encodes that $\varphi$ has a model of size $n$, which are hence contradictions. Following Riis, it is typical to encode the existence of the witnesses in longhand with a big disjunction, that we designate the *unary encoding*. As recently investigated in the works [19, 12, 13, 27, 22], it may also be possible to encode the existence of such witnesses *succinctly* by the use of a *binary encoding*. Essentially, the existence of the witness is now given implicitly as any propositional assignment to the relevant variables

34th Computational Complexity Conference (CCC 2019).
Editor: Amir Shpilka; Article No. 6; pp. 6:1–6:25


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

gives a witness, whereas in the unary encoding a solitary true literal tells us which is the witness. Combinatorial principles encoded in binary are interesting to study since, loosely speaking, they still preserve the hardness of the combinatorial principle encoded while giving a more succinct propositional representation. In certain cases this leads to obtain significant lower bounds in an easier way than for the unary case [19, 13, 27].

The central thrust of this work is to contrast the proof complexity (size) between the unary and binary encodings of natural combinatorial principles. This has not previously been done systematically in Proof Complexity, though it has been better-studied the in the "dual" area of SAT-solving [26, 29]. In the SAT community it is well-known one may try various different encodings of the 1-from-$n$ constraint to speed-up proofs of unsatisfiability as well as satisfiability. In [29, 37], what we call the binary encoding is referred to as *logarithmic*. The Pigeonhole Principle is explored experimentally in both [26] and Chapter 7 in [29, 37] (though sadly the binary encoding is not among the tests).

A principal motivation 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 $\mathsf{Res}(s)$ for the corresponding families of contradictions where witnesses are given in the binary encodings. $\mathsf{Res}(s)$, is a refutational proof system extending Resolution to $s$-bounded DNFs, introduced by Krajíček in [23]. Our approach is justified by observing that (see Lemma 26), for a family of contradictions encoding a principle which is expressible as $\Pi_2$ first-order formulae having no finite models, short $\mathsf{Res}(\log n)$ refutations of their *binary* encoding can be obtained from short Resolution refutations for the *unary* encoding. Lower bounds for $\mathsf{Res}(s)$ have appeared variously in the literature. Of most interest to us are those for the (moderately weak) Pigeonhole Principle $\mathsf{PHP}^{2n}_n$, for $\mathsf{Res}(\sqrt{\log n/\log\log n})$ in [35], improved to $\mathsf{Res}(\epsilon \log n/\log\log n)$ in [2]. A hierarchy in $\mathsf{Res}(s)$ is uncovered by the use of relativising the (Linear) Ordering Principle in [17].

Our first interest is the *$k$-Clique Principle*, whose precise Resolution complexity is still unknown; but we also study other principles, to make progress in the direction of our approach. The three combinatorial principles we deal with in this paper are: (1) the $k$-Clique Formulae, $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$; (2) the (weak) Pigeonhole Principle $\mathsf{PHP}^m_n$; and (3) the (Linear) Ordering Principle, $(\mathsf{L})\mathsf{OP}_n$. The *$k$-Clique Formulae* introduced in [10, 11, 6] are formulae 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, $\mathsf{PHP}^m_n$ is well-studied in proof complexity (see among others: [21, 35, 16, 31, 33, 32, 8, 15, 9, 7, 5, 3, 28]). The $\mathsf{LOP_n}$ formulae 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 [24, 36, 14].

## 1.1 *k*-Clique Principle

Deciding whether a graph has a $k$-clique it is one of the central problems in Computer Science and 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 $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ (respectively any CNF $F$), can be thought 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 $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ would then answer the above question for algorithms based on Resolution primitives. This

question was posed in [10], where it was also answered in the case of refutations in the form of trees (treelike Resolution). Recently in a major breakthrough Atserias et al. in [4] prove the $n^{\Omega(k)}$ lower bound for the case of read-once proofs (Regular resolution). The graph $G$ considered in [10, 4] to plug in the formula $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ to make it unsatisfiable was a random graph obtained by a slight variation of Erdös-Rényi distribution of random graphs as defined in [10]. But the exact Resolution complexity of $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$, for $G$ random is unknown. In the work [27], Lauria et al. consider the binary encoding of Ramsey-type propositional statements, having as a special case a binary version of $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$: $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$. They obtain optimal lower bounds for $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ in Resolution, which is $\mathsf{Res}(1)$.

Our main result (Theorem 7) is a $n^{\Omega(k)}$ lower bound for the size of refutations of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ in $\mathsf{Res}(o(\log\log n))$, when $G$ is a random graph as that defined in [10]. Lemma 2 in Section 3 proves that a lower bound in $\mathsf{Res}(\log)$ for the $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ would prove a lower bound in Resolution for $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$.

## 1.2 Weak Pigeonhole Principle

An interesting example to test the relative hardness of binary versions of combinatorial principle comes from the (weak) Pigeonhole Principle. In Section 4, we consider its binary version $\mathsf{Bin\text{-}PHP}^m_n$ and we prove that in $\mathsf{Res}(s)$, for all $\epsilon > 0$ and $s \leq \log^{\frac{1}{2-\epsilon}}(n)$, the shortest proofs of the $\mathsf{Bin\text{-}PHP}^m_n$, require size $2^{n^{1-\delta}}$, for any $\delta > 0$ (Theorem 22). This is the first size lower bound known for the $\mathsf{Bin\text{-}PHP}^m_n$ in $\mathsf{Res}(s)$. As a by-product of this lower bound we prove a lower bound of the order $2^{\Omega(\frac{n}{\log n})}$ (Theorem 18) for the size of the shortest Resolution refutation of $\mathsf{Bin\text{-}PHP}^m_n$. Our lower bound for $\mathsf{Res}(s)$ is obtained through a technique that merges together the random restriction method, an inductive argument on the $s$ of $\mathsf{Res}(s)$ and the notion of *minimal covering* of a $k$-DNF of [35]. Since we are not using any (even weak) form of Switching Lemma (as for instance in [35, 1]), we consider how tight is our lower bound in $\mathsf{Res}(s)$. We prove that $\mathsf{Bin\text{-}PHP}^m_n$ (Theorem 23) can be refuted in size $2^{O(n)}$ in treelike $\mathsf{Res}(1)$. Our upper bound is contrasting with the unary case of the Pigeonhole Principle, $\mathsf{PHP}^m_n$, which instead requires treelike $\mathsf{Res}(1)$ refutations of size $2^{\Omega(n\log n)}$, as proved in [9, 16].

For the Pigeonhole Principle, similarly to the $k$-Clique Principle, we can prove that short $\mathsf{Res}(\log n)$ refutations for $\mathsf{Bin\text{-}PHP}^m_n$ can be efficiently obtained from short $\mathsf{Res}(1)$ of $\mathsf{PHP}^m_n$ (Lemma 15). This allows us to prove that our lower bound is almost optimal: Buss and Pitassi, in [15], proved an upper bound of $2^{O(\sqrt{n\log n})}$ for the size of refuting $\mathsf{PHP}^m_n$ in $\mathsf{Res}(1)$ when $m \geq 2^{\sqrt{n\log n}}$, which by our Lemma 15 holds also for $\mathsf{Res}(\log n)$ proofs of $\mathsf{Bin\text{-}PHP}^m_n$. It follows that our exponential lower bound for $\mathsf{Bin\text{-}PHP}^m_n$ (Theorem 18) for any $m > n$ in $\mathsf{Res}(\log^{1/2-\epsilon} n)$ is almost optimal.

## 1.3 Contrasting unary and binary principles

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 [34] we consider principles which are expressible as first-order formulae with no finite model in $\Pi_2$-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 the *Ordering Principle*, which states that a finite partial order has a maximal element is one such principle. Its negation can be expressed in $\Pi_2$-form as: $\forall x,y,z\exists w\ \neg R(x,x) \wedge (R(x,y) \wedge R(y,z) \rightarrow R(x,z)) \wedge R(x,w)$. In Definition 25 we explain how to generate a binary encoding $\mathsf{Bin\text{-}C_n}$ from any combinatorial principle $\mathsf{C_n}$ expressible as a first order formulae in $\Pi_2$-form with no finite models and whose unary encoding we denote by $\mathsf{Un\text{-}C_n}$. Another example is the

Pigeonhole Principle whose negation of its relational form can be expressed as a $\Pi_2$-formula as $\forall x, y, z \exists w \neg R(x, 0) \wedge (R(x, z) \wedge R(y, z) \rightarrow x = y) \wedge R(x, w)$. Notice that in the case of the 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. This accounts for why, in the case of the Pigeonhole Principle, one can give another more efficient (in terms of number of variables) binary encoding (see Section 5 for details), than the one given by $\mathsf{Bin\text{-}C_n}$ applied to the $\mathsf{PHP}$. Nevertheless in Lemma 27 we observe that in Resolution efficient refutations for one encoding can be obtained from refutations of the other encoding. We propose a framework to compare lower bounds for the $\mathsf{Bin\text{-}C_n}$ in $\mathsf{Res}(s)$ with lower bounds for $\mathsf{Un\text{-}C_n}$ in $\mathsf{Res}(1)$ by proving in Lemma 26 that short Resolution refutations for $\mathsf{Un\text{-}C_n}$ produces short $\mathsf{Res}(\log n)$ refutations for $\mathsf{Bin\text{-}C_n}$.

### 1.3.1 Linear Ordering Principles

Linear ordering formulae $\mathsf{LOP_n}$ encodes a Linear Ordering Principle. They were used in [14, 20] as families of formulae witnessing the optimality of the size-width tradeoffs for Resolution ([8]), so that they require high width to be refuted, but still admit polynomial size refutations in Resolution. Here we face the following open question: is the binary encoding of $\mathsf{LOP_n}$ formula still efficiently refutable in Resolution? In answering this question we will show something stronger, as we study under what conditions the complexity of proofs in Resolution will not increase significantly (by more than a polynomial factor) when shifting from the unary encoding to the binary encoding. In Lemma 24 we prove that this is true for the negation of principles expressible as first order formula in $\Pi_2$-form involving *total variable comparisons*. Hence in particular the binary version of the Linear Ordering Principle $\mathsf{Bin\text{-}LOP_n}$. Finally, we also prove that the binary encoding of the Ordering Principle $\mathsf{Bin\text{-}OP_n}$, where antisymmetry is not encoded and hence there is no total variable comparison, is also polynomially provable in Resolution. Broadly speaking, these results are saying that shifting to the binary encodings is not destroying the hardness of a unary principle when working in Resolution. Hence binary encodings of combinatorial principles are meaningful benchmarks for Resolution to prove lower bounds for.

### 1.3.2 Binary encodings of principles versus their Unary functional encodings

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. This 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 5.3, 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 has complexity $2^{\Theta(n \log n)}$ [9, 16], while, as we prove in Subsection 4.1 (Theorem 23), the binary (functional) encoding is $2^{\Theta(n)}$. The unary encoding complexity is noted in [17] and remains true for the unary functional encoding with the same lower-bound proof. The binary encoding complexity is addressed directly in this paper.

### 1.4 Techniques and Organization

The method of random restrictions in Proof Complexity is often employed to prove size lower bounds. Loosely speaking the method works as follows: we consider formulae having a given specific combinatorial property $P$; after hitting, with a suitable random partial assignment, on an allegedly short proof of the formula we are refuting, we are left to prove that with high probability a formula with property $P$ is killed away from the proof. The growth rate as the probability approaches to 1 together with a counting argument using averaging (as the union bound), implies a lower bound on the number of formulae with property P in the proof. Lower bounds in $\mathsf{Res}(s)$ using random restrictions were known only for $s = 2$ (see [5]). Using a weak form of the Switching Lemma, lower bounds for $\mathsf{Res}(s)$ were obtained in [35, 1]. From the latter paper we use the notion of *covering number* of a $k$-DNF $F$, i.e. the minimal size of a set of variables to hit all the $k$-terms in $F$. In this work we merge the covering number with the random restriction method together with an inductive argument on the $s$, to get size lower bounds in $\mathsf{Res}(s)$ specifically for binary encoding of combinatorial principles.

After a section with the preliminaries, the paper is divided into four sections: one with the lower bound for the $k$-Clique Principle, one containing all the results for the (weak) Pigeonhole Principle, one for the contrasting the proof complexity between unary and binary principles containing all the results about the various Ordering Principles, and finally the last section containing a general approach to unary vs binary encodings for principle expressible as a $\Pi_2$-formulae.

## 2 Preliminaries

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 $k$ literals. A *clause* with $s$ literals is a $s$-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 $\bot$. A *$s$-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 $((v_1 \vee \neg v_2) \wedge (v_2 \vee v_3) \wedge (\neg v_1 \vee v_3))$ is an example of a 2-CNF.

We can now describe the propositional refutation system $\mathsf{Res}(s)$ ([23]). 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 $\mathsf{Res}(s)$ 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 the 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 "$\mathsf{Res}(s)$" could depend on $n$ – an important special case is $\mathsf{Res}(\log n)$.

Clearly, $\mathsf{Res}(1)$ 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 $\mathsf{Res}(1)$ refutation $\pi$ of $\mathcal{C}$ the width of $\pi$, $w(\pi)$ is the maximal width of a clause in $\pi$. The width refuting $\mathcal{C}$ in Res(1), $w(\vdash \mathcal{C})$, is the minimal width over all $\mathsf{Res}(1)$ refutations of $\mathcal{C}$.

A *covering set* for a $s$-DNF $\mathcal{D}$ is a set of literals $L$ such that each term of $\mathcal{D}$ has at least a literal in $L$. The *covering number* $c(\mathcal{D})$ of a $s$-DNF $\mathcal{D}$ is the minimal size of a covering set for $\mathcal{D}$.

Let $\mathcal{F}(x_1 \dots, x_n)$ be a boolean $s$-DNF (resp. $s$-CNF) defined over variables $X = \{x_1, \dots, x_n\}$. A *partial assignment* $\rho$ to $\mathcal{F}$ is a truth-value assignment to some of the variables of $\mathcal{F}$: $dom(\rho) \subseteq X$. By $\mathcal{F}\restriction_\rho$ we denote the formula $\mathcal{F}'$ over variables in $X \setminus dom(\rho)$ obtained from $\mathcal{F}$ after simplifying in it the variables in $dom(\rho)$ according to the usual boolean simplification rules of clauses and terms.

## 2.1 $\mathsf{Res}(s)$ vs Resolution

Similarly to what was done for treelike $\mathsf{Res}(s)$ refutations in [18], if we turn a Res $(s)$ refutation of a given set of $s$-clauses $\Sigma$ 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 are as follows.

Each vertex is labelled by a $s$-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 [30]). Obviously, the (only) source is labelled with the constant $\top$. There are two kinds of queries, which can be made by a vertex:

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

$$\begin{array}{ccc} & \mathcal{C} & \\ & ?\bigvee_{j\in J} l_j & \\ \top\swarrow & & \searrow\bot \\ \mathcal{C}\wedge\bigvee_{j\in J} l_j & & \mathcal{C}\wedge\bigwedge_{j\in J}\neg l_j \end{array} \tag{1}$$

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

$$\begin{array}{ccc} & \mathcal{C}\wedge\bigvee_{j\in J_1\cup J_2} l_j & \\ & ?\bigvee_{j\in J_1} l_j & \\ \top\swarrow & & \searrow\bot \\ \mathcal{C}\wedge\bigvee_{j\in J_1} l_j & & \mathcal{C}\wedge\bigvee_{j\in J_2} l_j \end{array} \tag{2}$$

There are two ways of forgetting information,

$$\begin{array}{c} \mathcal{C}_1\wedge\mathcal{C}_2 \\ \downarrow \\ \mathcal{C}_1 \end{array} \quad \text{and} \quad \begin{array}{c} \mathcal{C}\wedge\bigvee_{j\in J_1} l_j \\ \downarrow \\ \mathcal{C}\wedge\bigvee_{j\in J_1\cup J_2} l_j \end{array}, \tag{3}$$

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. A sink of the branching $s$-program must be labelled with the negation of a $s$-clause from $\Sigma$. Thus the branching $s$-program is supposed by default to solve the *Search problem for* $\Sigma$: given an assignment of the variables, find a clause which is falsified under this assignment.

The equivalence between a Res $(s)$ refutation of $\Sigma$ 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.

Finally, we mention that the queries of the form (1) and (2) as well as forget-rules of the form (3) give rise to a Prover-Adversary game (see [30] where this game was introduced for Resolution). In short, Adversary claims that $\Sigma$ 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 Res $(k)$ refutation of $\Sigma$.

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

**Proof.** In the branching program, consider a $(k+1)$-CNF record $\phi$ whose covering number $< d$ is witnessed by variable set $V' := \{v_1, \ldots, v_d\}$. 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 record $\phi$ in our original branching program we expand a mini-tree of size $2^{d+2}$ with $2^{d+1}$ leaves questioning all the variables of $V'$ as well as the literal $l_{k+1}$. Clearly, each evaluation of these reduces $\phi$ to a $k$-CNF that logically implies $\phi$. 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 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 formulae $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ for $G$, has variables $v_{i,q}$ with $i \in [k], a \in [n]$, 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 $\mathcal{G}^n_k$ 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.

$\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ 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 is explained in Section 5.2. Let $a \in [n]$ and let $a_1 \ldots a_{\log n}$ be its binary representation. Each (unary) variable $v_{i,j}$ semantically corresponds to the conjunction $(\omega_{i,1}^{a_1} \wedge \ldots \wedge \omega_{i,\log n}^{a_{\log n}})$, where

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

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

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

The wide clauses from the unary encoding simply disappear in the binary encoding being implicit.

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

▶ **Lemma 2.** *Suppose there are Resolution refutations of* $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ *of size* $S$*. Then there are* $\mathsf{Res}(\log n)$ *refutations of* $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ *of size* $S$*.*

**Proof.** Where the decision DAG for $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ 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 $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$, and the result follows. ◀

Following [10, 4, 27] we consider $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ formulae where $G$ is a random graph distributed according to a variation of the Erdös-Rényi as defined in [10]. In the standard model, random graphs on $n$ vertices are constructed by including every edge independently with probability $p$. It is known that $k$-cliques appear at the threshold probability $p^* = n^{-\frac{2}{k-1}}$. If $p < p^*$, then with high probability there is no $k$-clique. By $\mathcal{G}^n_{k,\epsilon}(p)$ we denote the distribution on random multipartite Erdős-Renyi graph with $k$ blocks $V_i$ of $n$ vertices each, where each edge is present with probability $p$ depending on $\epsilon$. For $p = n^{-(1+\epsilon)\frac{2}{k-1}}$ we just write $\mathcal{G}^n_{k,\epsilon}$.

We use the notation $G = (\bigcup_{b\in[k]} V_b, E) \sim \mathcal{G}^n_k(p)$ to say that $G$ is a graph drawn at random from the distribution $\mathcal{G}^n_k(p)$.

In the next sections we explore lower bounds for $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ in $\mathrm{Res}(s)$ for $s \geq 1$, when $G \sim \mathcal{G}^n_k(p)$.

## 3.1 Isolating the properties of $G$

Let $\alpha$ be a constant such that $0 < \alpha < 1$. Define a set of vertices $U$ in $G$, $U \subseteq V$ to be an *$\alpha$-transversal* if: (1) $|U| \leq \alpha k$, 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 *extendible* in a block $b \in \overline{B(U)}$ if there exists a vertex $a \in V_b$ which 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 $U$ i.e. $N_c(U) = \{v \in V \mid v \in \bigcap_{u\in U} N(u)\}$.

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

▶ **Definition 3.** *Let* $0 < \alpha, \beta < 1$*. A* $\alpha$*-transversal set of vertices* $U$ *is* $\beta$*-extendible, if for all* $\beta$*-total restriction* $\sigma$*, there is a node* $v^b$ *in each block* $b \in \overline{B(U)}$*, such that* $\sigma$ *is consistent with* $v^b$*.*

▶ **Lemma 4** (Extension Lemma). *Let $0 < \epsilon < 1$, let $k \leq \log n$. Let $1 > \alpha > 0$ and $1 > \beta > 0$ such that $1 - \beta > \alpha(2 + \epsilon)$. Let $G \sim \mathcal{G}^n_{k,\epsilon}$. With high probability both the following properties hold:*

1. *all $\alpha$-transversal sets $U$ are $\beta$-extendible;*
2. *$\mathcal{G}$ does not have a $k$-clique.*

**Proof.** Let $U$ be an $\alpha$-transversal set and $\sigma$ be a $\beta$-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 $\sigma$ is applied, in each block $b \in \overline{B(U)}$ remain $2^{\log n - \beta \log n} = n^{1-\beta}$ available vertices. Hence the probability that we cannot extend $U$ in each block of $\overline{B(U)}$ after $\sigma$ is applied is $(1 - p^{\alpha k})^{n^{1-\beta}}$. Fix $c = 2 + \epsilon$ and $\delta = 1 - \beta - \alpha c$. Notice that $\delta > 0$ by our choice of $\alpha$ and $\beta$. Since $p = \frac{1}{n^{\frac{c}{k}}}$, previous probability is $(1 - 1/n^{\alpha c})^{n^{1-\beta}}$, which is asymptotically $e^{-\frac{n^{1-\beta}}{n^{\alpha c}}} = e^{-n^{\delta}}$ .

There are $\binom{k}{\alpha k}$ possible $\alpha$-transversal sets $U$ and $\binom{\log n}{\beta \log n} \cdot k$ possible $\beta$-total restrictions $\sigma$.

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

Notice that the last inequality holds since $k \leq \log n$. Hence the probability that there is in $G$ no $\alpha$-transeversal set $U$ which is $\beta$-extendible is going to 0 as $n$ grows.

To bound the probability that $\mathcal{G}$ contains a $k$-clique, notice that the expected number of $k$ cliques is $\binom{n}{k} \cdot p^{\binom{k}{2}} \leq n^k \cdot p^{(k(k-1)/2)}$. Recalling $p = 1/n^{c/k}$, we get that the probability that $G$ does not have a $k$-clique is $n^k \cdot n^{-c(k-1)/2} = n^{k - c(k-1)/2}$. Since $c = 2 + \epsilon$, $k - c(k-1)/2 = 1 - \frac{\epsilon}{2}(k-1)$. Hence $n^k \cdot n^{-c(k-1)/2} \leq 2^{-\log n}$ for sufficiently large $n$ and since $k \leq \log n$.

So the probability that either property (1) or (2) does not hold is bounded above by $2^{\log^2 n} \cdot e^{-n^{\delta}} + 2^{-\log^2 n}$ which is below 1 for sufficiently large $n$. ◀

## 3.2 Res($s$) lower bounds for Bin-Clique$^{\mathsf{n}}_{\mathsf{k}}$

Let $s \geq 1$ be an integer. Call a $\frac{1}{2^{s+1}}$-total assignment to the variables of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ an *$s$-restriction*. A *random $s$-restriction* for $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ 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 $G$ be graph over $nk$ nodes and $k$ blocks which does not contain a $k$-clique. Fix $\delta = \frac{1}{24^2}$ and $\mathsf{p}(s) = 2^{(s+1)^2}$ and $\mathsf{d}(s) = (\mathsf{p}(s)s)^s$.

Consider the following property.

▶ **Definition 5** (Property $\mathsf{Clique}(G, s, k)$). *For any $\gamma \geq 2$ and for any $\gamma$-restriction $\rho$, there are no Res(s) refutations of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_\rho$ of size less than $n^{\frac{\delta(k-1)}{\mathsf{d}(s)}}$.*

If property $\mathsf{Clique}(G, s, k)$ holds, we immediately have $n^{\Omega(k)}$ size lower bounds for refuting $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ in $Res(s)$.

▶ **Corollary 6.** *Let $s, k$ be integers, $s \geq 1, k > 1$. Let $G$ be a graph and assume that $\mathsf{Clique}(G, s, k)$ holds. Then there are no Res(s) refutations of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ of size smaller that $n^{\delta \frac{k-1}{\mathsf{d}(s)}}$.*

**Proof.** Choose $\rho$ to be any $s$-restriction, for $\gamma \geq 1$. 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 $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ in $\mathsf{Res}(s)$ as long as $s = o(\log\log n)$.

▶ **Theorem 7.** *Let $0 < \epsilon < 1$ be given. Let $k$ be an integer with $k > 1$. Let $s$ be an integer with $1 < s \leq \frac{1}{2}\log\log n$. Then there exists a graph $G$ such that Res(s) refutations of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ have size $n^{\Omega(k)}$.*

**Proof.** Let $1 > \alpha > 0$ and $1 > \beta > 0$ such that $1 - \beta > \alpha(2 + \epsilon)$. By Lemma 4, we can fix $G \sim \mathcal{G}^n_{k,\epsilon}$ such that:

1. all $\alpha$-transversal sets $U$ are $\beta$-extendible;
2. $\mathcal{G}$ does not have a $k$-clique.

We will prove, by induction on $s = o(\log\log n)$, that property $\mathsf{Clique}(G, s, k)$ does hold. The result then follows by Corollary 6. Lemma 8 is the base case and Lemma 9 the inductive case. ◀

▶ **Lemma 8** (Base Case)**.** $\mathsf{Clique}(G, 1, k)$ *does hold.*

**Proof.** Fix $\beta = \frac{3}{4}$ and $\alpha = \frac{1}{4(2+\epsilon)} \geq \frac{1}{12}$. Notice that $\mathsf{d}(1) = 16$. Let $\rho$ be a 1-restriction, that is a $\frac{1}{4}$-total assignment. We claim that any Resolution refutation of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_\rho$ must have width at least $\frac{k\log n}{24}$. This is a consequence of 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. Since $\frac{1}{4} + \frac{1}{2} = \frac{3}{4}$, this allows the game to continue until some record has width at least $\frac{\log n}{2} \cdot \frac{k}{12} = \frac{k\log n}{24}$. Size-width tradeoffs for Resolution [8] tells us that minimal size to refute any unsat CNF $F$ is lower bounded by $2^{\frac{(w(\vdash F) - w(F))^2}{16V(F)}}$[1]. In our case $w(F) = 2\log n$ and $V(F) = k\log n$, hence the minimal size required is $\geq 2^{\frac{(\frac{k\log n}{24} - 2\log n)^2}{16k\log n}} = 2^{\frac{\log n(\frac{k}{24}-2)^2}{16k}} = n^{\frac{(\frac{k}{24}-2)^2}{16k}}$. It is not difficult to see that $\frac{(\frac{k}{24}-2)^2}{16k} \geq \frac{(k-1)}{16\cdot 24^2}$. Since $\delta = \frac{1}{24^2}$ and $\mathsf{d}(1) = 16$ the result is proved. ◀

▶ **Lemma 9** (Inductive Case)**.** $\mathsf{Clique}(G, s-1, k)$ *implies* $\mathsf{Clique}(G, s, k)$.

**Proof.** Recall that we fixed $\mathsf{p}(s) = 2^{(s+1)^2}$ and $\mathsf{d}(s) = (\mathsf{p}(s)s)^s$. Set $L(s) = n^{\frac{\delta(k-1)}{\mathsf{d}(s)}}$ and $\chi(s) = \frac{(s-1)^{s-1}}{s^s 2^{3s^2+s}}$. (Proof of the next claim is postponed after the proof.)

▷ Claim 10. $\ln L(s) = \chi(s)\ln L(s-1)$

We prove the contrapositive of the statement of the Lemma. Assume there is some $s$-restriction $\rho$ such that there exists a Res($s$) refutation $\pi$ of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_\rho$ with size less than $L(s)$. We prove that that there is a $(s-1)$-restriction $\tau$ such that there are $\mathsf{Res}(s-1)$ proofs of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_\tau$ of size $< L(s-1)$.

Consider the function:

$$f(s,n) = \frac{(1-\chi(s))}{(\ln 2)\,\mathsf{d}(s-1)} - \frac{4}{\delta(k-1)\ln n}.$$

$f(s,n)$ is lower bounded as follows (see the proof after the the proof of this Lemma).

▷ Claim 11. For sufficiently large $n$ and for all $s \geq 2$,

$$f(s,n) > \frac{1}{(\mathsf{p}(s)s)^{s-1}}.$$

[1] According to [25] Th 8.11

Fix the covering number as:

$c = f(s,n)\delta(k-1)\ln n$

Define $r = \frac{c}{s}$ and let us call a *bottleneck* a record $R$ in $\pi$ whose covering number is $\geq c$. Hence in such a 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 the terms of the $s$-DNF forming the record $R$.

Let $\sigma$ be a *$s$-random restriction* on the variables of $\mathsf{Bin\text{-}Clique}_{\mathsf{k}}^{\mathsf{n}}(G)\restriction_\rho$. Let us say that $\sigma$ *kills a tuple* $T$ if it sets to 0 all literals in $T$ (notice that a record is the negation of a $s$-DNF) and that $T$ *survives* $\sigma$ otherwise. And that $\sigma$ *kills* $R$ if it kills at at least one of the tuples in $R$. Let $\Sigma_i$ be the event that $T_i$ survives $\sigma$ and $\Sigma_R$ the event that $R$ survives $\sigma$. We claim (postponing the proof) that

▷ Claim 12. $\Pr[\Sigma_R] \leq (1 - \frac{1}{\mathsf{p}(s)})^r$.

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

$$\Pr[\exists R \in \pi\restriction_\tau \colon \Sigma_R] \leq |\pi\restriction_\tau|(1 - \frac{1}{\mathsf{p}(s)})^r.$$

We claim that this probability is $< 1$. Notice that $(1 - \frac{1}{\mathsf{p}(s)})^r \leq e^{-\frac{c}{s\,\mathsf{p}(s)}}$ using the definition of $r$. So to prove the claim it is sufficient to prove that $|\pi\restriction_\tau| < e^{\frac{c}{\mathsf{p}(s)s}}$ or equivalently that $\ln|\pi\restriction_\tau| < \frac{c}{s\,\mathsf{p}(s)}$. But $\ln|\pi\restriction_\tau| \leq \ln|\pi| = \ln L(s) = \frac{1}{s\,\mathsf{p}(s)}\frac{\delta(k-1)\ln n}{(\mathsf{p}(s)s)^{s-1}}$. Since by Claim 11 $f(s,n) > \frac{1}{(\mathsf{p}(s)s)^{s-1}}$, then $\ln|\pi\restriction_\tau| < \frac{f(s,n)\delta(k-1)\ln n}{s\,\mathsf{p}(s)} = \frac{c}{s\,\mathsf{p}(s)}$, where the last inequality follows by definition of $c$.

Since in $\pi\restriction_\tau$ there is no bottleneck, by Lemma 1, we can morph $\pi\restriction_\tau$ through the restriction $\tau$ to a $\mathsf{Res}(s-1)$ refutation of $\mathsf{Bin\text{-}Clique}_{\mathsf{k}}^{\mathsf{n}}(G)\restriction_\tau$ of size $2^{c+2}\cdot L(s)$. Hence the Lemma is proved arguing that

$$2^{c+2}\cdot L(s) < L(s-1) \tag{4}$$

Since by Claim 10, $\ln L(s) = \chi(s)\ln L(s-1)$, we have the following equivalences:

$$(c+2)\ln 2 + \ln L(s) < \ln L(s-1) \qquad \text{Passing to ln of Eq. 4} \tag{5}$$

$$(c+2)\ln 2 < \ln L(s-1)(1-\chi(s)) \tag{6}$$

$$(f(s,n)\delta(k-1)\ln n + 2)\ln 2 < \frac{\delta(k-1)\ln n}{\mathsf{d}(s-1)}\cdot\frac{(1-\chi(s))}{} \qquad \text{def of } c \text{ and of } L(s-1) \tag{7}$$

$$f(s,n)\delta(k-1)\ln n + 2 < \frac{\delta(k-1)\ln n}{\mathsf{d}(s-1)}\cdot\frac{(1-\chi(s))}{\ln 2} \qquad \text{dividing by } \ln 2 \tag{8}$$

$$f(s,n)\delta(k-1)\ln n < \frac{\delta(k-1)\ln n}{\mathsf{d}(s-1)}\cdot\frac{(1-\chi(s))}{\ln 2} - 2 \qquad \text{subtracting 2} \tag{9}$$

$$f(s,n) < \frac{(1-\chi(s))}{(\ln 2)\,\mathsf{d}(s-1)} - \frac{2}{\delta(k-1)\ln n}. \qquad \text{dividing by } \delta(k-1)\ln n \tag{10}$$

The last line is true since by its definition $f(s,n) = \frac{(1-\chi(s))}{(\ln 2)\,\mathsf{d}(s-1)} - \frac{4}{\delta(k-1)\ln n}$. ◀

Notice that the due to the definition of $L(s)$ the proof can be carried as long as $(s\,\mathsf{p}(s))^s \leq \ln n$ which means $s = o(\log\log n)$.

Proof of Claim 10. Notice that $\mathsf{p}(s-1) = 2^{s^2}$ and that $\mathsf{p}(s) = 2^{s^2}2^{2s+1} = \mathsf{p}(s-1)2^{2s+1}$. Consider the following equalities

$$\ln L(s) = \frac{\delta(k-1)\ln n}{(\mathsf{p}(s)s)^s} \tag{11}$$

$$= \frac{\delta(k-1)\ln n}{(\mathsf{p}(s-1)2^{2s+1})^s s^s} \cdot \frac{(s-1)^{s-1}}{(s-1)^{s-1}} \tag{12}$$

$$= \frac{\delta(k-1)\ln n}{\mathsf{p}(s-1)^{s-1}(s-1)^{s-1}} \cdot \frac{(s-1)^{s-1}}{s^s\,\mathsf{p}(s-1)(2^{2s+1})^s} \tag{13}$$

$$= \frac{\delta(k-1)\ln n}{\mathsf{p}(s-1)^{s-1}(s-1)^{s-1}} \cdot \frac{(s-1)^{s-1}}{s^s\,\mathsf{p}(s-1)2^{2s^2+s}} \tag{14}$$

$$= \frac{\delta(k-1)\ln n}{\mathsf{d}(s-1)} \cdot \frac{(s-1)^{s-1}}{s^s 2^{s^2}2^{2s^2+s}} \tag{15}$$

$$= L(s-1) \cdot \frac{(s-1)^{s-1}}{s^s 2^{3s^2+s}} \tag{16}$$

Notice that $\chi(s) = \frac{(s-1)^{s-1}}{s^s 2^{3s^2+s}}$ so the result follows. ◁

Proof of Claim 11. For $n \to \infty$, $\frac{4}{\delta(k-1)\ln n} \to 0$, so for a sufficiently large $n$ we can ignore the term $\frac{4}{\delta(k-1)\ln n}$. Moreover since $\ln 2 < 1$ we forgot the factor $\frac{1}{\ln 2}$ in $f(s,n)$. We have to show that for all $s \geq 2$

$$\frac{(1-\chi(s))}{(\mathsf{p}(s-1)(s-1))^{s-1}} > \frac{1}{(\mathsf{p}(s)s)^{s-1}}. \tag{17}$$

First we bound the RHS in a convenient form. First since $\frac{1}{s-1} > \frac{1}{s}$ the claim in Eq 17 follows from proving that

$$\frac{(1-\chi(s))}{(\mathsf{p}(s-1)(s-1))^{s-1}} > \frac{1}{(\mathsf{p}(s)(s-1))^{s-1}}. \tag{18}$$

Recall from proof of Claim 10 that $\mathsf{p}(s) = \mathsf{p}(s-1)2^{2s+1}$. Hence we can write the denominator $(\mathsf{p}(s)(s-1))^{s-1}$ of RHS of Eq. 18 as

$$(\mathsf{p}(s)(s-1))^{s-1} = (\mathsf{p}(s-1)(s-1))^{s-1} \cdot (2^{2s+1})^{s-1} \tag{19}$$

$$= (\mathsf{p}(s-1)(s-1))^{s-1} \cdot 2^{2s^2-(s+1)} \tag{20}$$

Hence Eq. 18 follows from proving

$$\frac{(1-\chi(s))}{(\mathsf{p}(s-1)(s-1))^{s-1}} > \frac{1}{(\mathsf{p}(s-1)(s-1))^{s-1} \cdot 2^{2s^2-(s+1)}} \tag{21}$$

Multiplying both sides by $(\mathsf{p}(s-1)(s-1))^{s-1}$ this is equivalent to prove that

$$(1-\chi(s)) > \frac{1}{2^{2s^2-(s+1)}} \tag{22}$$

Which is equivalent to prove that

$$(1-\chi(s)) > \frac{2^{s+1}}{2^{2s^2}} \tag{23}$$

Now we work on a more convenient form of LHS. Recall that

$$\chi(s) = \frac{(s-1)^{s-1}}{s^s 2^{3s^2+s}}$$

so that

$$1 - \chi(s) = \frac{s^s 2^{3s^2+s} - (s-1)^{s-1}}{s^s 2^{3s^2+s}} \tag{24}$$

So Eq 23 can be rewritten as

$$\frac{s^s 2^{3s^2+s} - (s-1)^{s-1}}{s^s 2^{3s^2+s}} > \frac{2^{s+1}}{2^{2s^2}} \tag{25}$$

Multiplying both sides by $s^s 2^{3s^2+s}$ we have the equivalent equation

$$s^s 2^{3s^2+s} - (s-1)^{s-1} > 2^{s+1} s^s 2^{s^2+s} \tag{26}$$

which, dividing both sides by $s^s$ is equivalent to prove

$$2^{3s^2+s} - \frac{(s-1)^{s-1}}{s^s} > 2^{s^2+2s+1} \tag{27}$$

First we claim that $\frac{(s-1)^{s-1}}{s^s} < 1$, which is equivalent tor prove that $(s-1)\ln(s-1) - s\ln(s) < 0$ by passing to logarithms. But $(s-1)\ln(s-1) - s\ln(s) < (s-1)\ln(s-1) - (s-1)\ln(s) < (s-1)\ln(s-1) - (s-1)\ln(s-1) = 0$.

So

$$2^{3s^2+s} - \frac{(s-1)^{s-1}}{s^s} > 2^{s^2+s} - 1$$

and Eq 27 follows from proving that

$$2^{3s^2+s} - 1 \geq 2^{s^2+2s+1} \tag{28}$$

divide both sides by the RHS, which is $2^{s^2+2s+1}$ so that we want to prove that

$$2^{3s^2+s-(s^2+2s+1)} - \frac{1}{2^{s^2+2s+1}} \geq 1 \tag{29}$$

Again $\frac{1}{2^{s^2+2s+1}} \leq 1$ and $2^{3s^2+s-(2s^2+2s+1)} = 2^{2s^2-(s+1)}$, so that Eq 29 follows from proving that

$$2^{2s^2-(s+1)} - 1 \geq 1 \tag{30}$$

$2^{2s^2-(s+1)}$ is a growing function in $s$ and for $s = 2$ is value is exactly $2^5 = 32 > 2$. Hence it is always true that $2^{s^2-(s+1)} \geq 2$, which proves Eq 29 and hence our Claim. ◁

Proof of Claim 12. Since $T_1, \ldots, T_r$ are tuples in $R$, then $\Pr[\Sigma_R] \leq \Pr[\Sigma_1 \wedge \ldots \wedge \Sigma_r]$. Moreover $\Pr[\Sigma_1 \wedge \ldots \wedge \Sigma_r] = \prod_{i=1}^{r} \Pr[\Sigma_i | \Sigma_1 \wedge \ldots \wedge \Sigma_{i-1}]$. We will prove that for all $i = 1, \ldots, r$,

$$\Pr[\Sigma_i | \Sigma_1 \wedge \ldots \wedge \Sigma_{i-1}] \leq \Pr[\Sigma_i] \tag{31}$$

Hence the result follows from Lemma 13 which is proving that $\Pr[\Sigma_i] \leq 1 - \frac{1}{\mathsf{p}(s)}$.

By Lemma 14 (i), to prove that Equation 31 holds, we show that $\Pr[\Sigma_i | \neg\Sigma_1 \vee \ldots \vee \neg\Sigma_{i-1}] \geq \Pr[\Sigma_i]$. We claim that for $j \in [r], i \neq j$:

$$\Pr[\Sigma_i | \neg\Sigma_j] \geq \Pr[\Sigma_i] \tag{32}$$

Hence repeated applications of Lemma 14 (ii), prove that $\Pr[\Sigma_i | \neg\Sigma_1 \vee \ldots \vee \neg\Sigma_{i-1}] \geq Pr[\Sigma_i]$.

To prove Equation 32, let $B(T_i)$ be the set of blocks mentioned in $T_i$. If $B(T_i)$ and $B(T_j)$ are disjoint, then clearly $\Pr[\Sigma_i | \neg\Sigma_j] = \Pr[\Sigma_i]$. When $B(T_i)$ and $B(T_j)$ are not disjoint, we reason as follows: For each $\ell \in B(T_i)$, let $T_i^\ell$ be the set of variables in $T_i$ mentioning block $\ell$. $T_i$ is hence partitioned into $\bigcup_{\ell \in B(T_i)} T_i^\ell$ and hence the event "$T_i$ surviving $\sigma$", can be partitioned into the sum of the events that $T_i^\ell$ survives to $\sigma$, for $\ell \in B(T_i)$. Denote by $\Sigma_i^\ell$ the event "$T_i^\ell$ survives $\sigma$" and let A=$B(T_i) \cap B(T_j)$ and $B = B(T_i) \setminus (B(T_i) \cap B(T_j))$. The following inequalities holds:

$$\begin{aligned}
\Pr[\Sigma_i | \neg\Sigma_j] &= \Pr[\exists \ell \in B(T_i) : \Sigma_i^\ell | \neg\Sigma_j] && (33)\\
&= \sum_{\ell \in B(T_i)} \Pr[\Sigma_i^\ell | \neg\Sigma_j] && (34)\\
&= \sum_{\ell \in A} \Pr[\Sigma_i^\ell | \neg\Sigma_j] + \sum_{\ell \in B} \Pr[\Sigma_i^\ell | \neg\Sigma_j] && (35)\\
& && (36)
\end{aligned}$$

Since $B$ is disjoint from $B(T_j)$, as for the case above for each $\ell \in B$, $\Pr[\Sigma_i^\ell | \neg\Sigma_j] = \Pr[\Sigma_i^\ell]$. Then:

$$\sum_{\ell \in B} \Pr[\Sigma_i^\ell | \neg\Sigma_j] = \sum_{\ell \in B} \Pr[\Sigma_i^\ell] \tag{37}$$

$$\tag{38}$$

Notice that $T_i$ and $T_j$ are disjoint, hence knowing that some indices in blocks $\ell \in A$ are already chosen to kill $T_j$, only increase the chances of $T_i$ to survive (since less positions are left in the blocks $\ell \in A$ to potentially kill $T_i$).

Hence:

$$\sum_{\ell \in A} \Pr[\Sigma_i^\ell | \neg\Sigma_j] \geq \sum_{\ell \in A} \Pr[\Sigma_i^\ell] \tag{39}$$

$$\tag{40}$$

Which proves the claim since:

$$\sum_{\ell \in A} \Pr[\Sigma_i^\ell] + \sum_{\ell \in B} \Pr[\Sigma_i^\ell] = \Pr[\Sigma_i] \tag{41}$$

◁

▶ **Lemma 13.** *Let $\rho$ be a $s$-random restriction. For all $s$-tuples $S$:*

$$\Pr[S \textit{ survives } \rho] \leq 1 - \frac{1}{\mathsf{p}(s)}$$

**Proof.** Let $T = (\ell_{i_1,j_1}, \ldots, \ell_{i_s,j_s})$ be an $s$-tuple made of of disjoint literals of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$. We say that $T$ is *perfect* if all literals are bits of a same block.

Let $\gamma = \frac{1}{2^{s+1}}$. A block with $r$ distinct bits contributes a factor of

$$\frac{\binom{\gamma \log n}{r}}{\binom{\log n}{r}} \cdot \frac{1}{2^r}$$

to the probability that the $s$-tuple **does not** survive. Expanding the left-hand part of this 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

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

The result now follows when we recall that the probability of surviving is maximised when the probability of not surviving is minimised. ◀

▶ **Lemma 14.** *Let $A, B, C$ three events such that $\Pr[A], \Pr[B], \Pr[C] > 0$:*

**(i)** *If $\Pr[A|\neg B] \geq \Pr[A]$ then $\Pr[A|B] \leq \Pr[A]$;*

**(ii)** *$\Pr[A|B] \geq \Pr[A]$ and $\Pr[A|C] \geq \Pr[A]$. Then $\Pr[A|B \vee C] \geq \Pr[A]$.*

**Proof.** For part (i) consider the following equivalences:

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

For part (ii) consider the following inequalities:

$$\begin{array}{lcl}
\Pr[A|B \vee C] & = & \frac{\Pr[A \wedge (B \vee C)]}{\Pr[B \vee C]} \\
& \geq & \frac{\Pr[A \wedge B]}{\Pr[B \vee C]} + \frac{\Pr[A \wedge C]}{\Pr[B \vee C]} \\
& = & \frac{\Pr[A \wedge B]}{\Pr[B]} \cdot \frac{\Pr[B]}{\Pr[B \vee C]} + \frac{\Pr[A \wedge C]}{\Pr[C]} \cdot \frac{\Pr[C]}{\Pr[B \vee C]} \\
& = & \Pr[A|B] \cdot \frac{\Pr[B]}{\Pr[B \vee C]} + \Pr[A|C] \cdot \frac{\Pr[C]}{\Pr[B \vee C]} \\
& \geq & \Pr[A] \cdot (\frac{\Pr[B] + \Pr[C]}{\Pr[B \vee C]}) \\
& \geq & \Pr[A]
\end{array}$$

◀

## 4 The weak Pigeonhole Principle

For $n < m$, let $\mathsf{Bin\text{-}PHP}^m_n$ be the binary encoding of the (weak) Pigeonhole Principle. $\mathsf{Bin\text{-}PHP}^m_n$ is a well-known formula and its definition can be found in Section 5. First notice that an analogous of Lemma 2 holds for the Pigeonhole Principle too.

▶ **Lemma 15.** *Suppose there are Resolution refutations of $\mathsf{PHP}^m_n$ of size $S$. Then there are $\mathsf{Res}(\log n)$ refutations of $\mathsf{Bin\text{-}PHP}^m_n$ of size $S$.*

Let $\rho$ be a partial assignment (a restriction) to the variables of $\mathsf{Bin\text{-}PHP}^m_n$. We call $\rho$ a *$t$-bit* restriction if $\rho$ 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 $a_1 \dots a_{\log n}$ be the binary representation of $a$. We say that a restriction $\rho$ 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 $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\rho}$, $\mathsf{Bin\text{-}PHP}^m_n$ restricted by $\rho$. We will also consider the situation in which an $s$-bit restriction is applied to some $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\rho}$, creating $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\tau}$, where $\tau$ is an $s+t$-bit restriction.

Throughout this section, let $u = u(n,t) := (\log n) - t$. We do not use this shorthand 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).

▶ **Lemma 16.** *Let $\rho$ be a $t$-bit restriction for $\mathsf{Bin\text{-}PHP}^m_n$. Any decision DAG for $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\rho}$ must contain a record which mentions $\frac{n}{2^t}$ pigeons.*

**Proof.** Let Adversary play in the following fashion. While some pigeon is not mentioned at all, 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 $\rho$), will Adversary forget the hole he assigned it.

It remains to argue that Adversary must force Prover to produce a record of width $\geq \frac{n}{2^{t+1}}$ and for this it suffices to argue that Adversary can remain consistent with $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\rho}$ up until the point that such a 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 $\rho$ has already assigned some bits).

The current 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 $\log n - t$ bits which corresponds to $\frac{n}{2^t}$ holes. Since we have assigned fewer than $\frac{n}{2^t}$ pigeons to holes, one of these must be available, and the result follows. ◀

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

▶ **Definition 17** (Property $\mathsf{PHP}(s,t)$)**.** *Let $s, t \geq 1$. For any $t$-bit restriction $\rho$ to $\mathsf{Bin\text{-}PHP}^m_n$, there are no $\mathsf{Res}(s)$ refutations of $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\rho}$ of size smaller than $e^{\frac{n}{4^{\xi(s)+1}s!2^t u^{\xi(s)}}}$.*

▶ **Theorem 18.** *Let $\rho$ be a $t$-bit restriction for $\mathsf{Bin\text{-}PHP}^m_n$. Any decision DAG for $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\rho}$ is of size $2^{\Omega(\frac{n}{\log n})}$ (indeed, asymptotically of size $\geq e^{\frac{n}{2^{t+2}u}}$).*

**Proof.** Call a *bottleneck* a record in the decision DAG that mentions $\frac{n}{2^{t+1}}$ 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

$$\left(\frac{u-1}{u} + \frac{1}{2u}\right)^{\frac{n}{2^{t+1}}} = \left(1 - \frac{1}{2u}\right)^{u \cdot \frac{n}{2^{t+1}u}} \leq \frac{1}{e^{\frac{n}{2^{t+2}u}}},$$

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 $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\rho}$. By the union bound there is a random restriction that kills all bottlenecks and this leaves a decision DAG for some $\mathsf{Bin\text{-}PHP}^m_n{\restriction_\sigma}$, where $\sigma$ is a $(t+1)$-bit restriction for $\mathsf{Bin\text{-}PHP}^m_n$. However, we know from Lemma 16 that such a refutation must involve a record mentioning $\frac{n}{2^{t+1}}$ pigeons. This is now the desired contradiction. ◀

Note that the previous theorem could have been proved, like Lemma 8, 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 19.** *Property* $\mathsf{PHP}(1,t)$ *holds, for each* $t < \log n$.

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

Let $(\ell_{i_1,j_1},\ldots,\ell_{i_s,j_s})$ be an $s$-tuple made of disjoint literals of $\mathsf{Bin\text{-}PHP}_n^m \restriction_\rho$. We say that a tuple is *perfect* if all literals come from the same pigeon.

▶ **Lemma 20.** *Let* $s$ *be an integer,* $s \geq 1$ *and* $s+t < \log n$. *Let* $\sigma$ *be a random* $s$*-bit restriction over* $\mathsf{Bin\text{-}PHP}_n^m \restriction_\rho$ *where* $\rho$ *is itself some* $t$*-bit restriction over* $\mathsf{Bin\text{-}PHP}_n^m$. *Let* $T$ *be a perfect* $s$*-tuple of* $\mathsf{Bin\text{-}PHP}_n^m \restriction_\rho$. *Then for all* $s$*-tuples* $S$*:*

$$\Pr[T \textit{ survives } \sigma] \geq \Pr[S \textit{ survives } \sigma].$$

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

**Proof.** A pigeon with $r$ distinct bits contributes to not surviving a factor of

$$\frac{s}{\log n\ -t} \cdot \frac{s-1}{\log n\ -t-1} \cdots \frac{s-r+1}{\log n - t - r + 1} \cdot \frac{1}{2^r}.$$

Noting that

$$\frac{s}{\log n\ -t} \cdot \frac{s-1}{\log n\ -t-1} \cdots \frac{1}{\log n - t - s + 1} \cdot \frac{1}{2^r} > \frac{1}{u^s}$$

the result now follows when we recall that the probability of surviving is maximised when the probability of not surviving is minimised. ◀

▶ **Theorem 21.** *Let* $s > 1$ *and* $s + t < \log n$. *Then,* $\mathsf{PHP}(s-1, s+t)$ *implies* $\mathsf{PHP}(s,t)$.

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

Call a *bottleneck* a record that has covering number $\geq \frac{n}{4^{\xi(s)} \cdot (s-1)! 2^t u^{\xi(s-1)}}$. In such a 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 $(\ell_1^1,\ldots,\ell_1^s),\ldots,(\ell_r^1,\ldots,\ell_r^s)$ so that each $s$-tuple is a clause in the record and no pigeon appearing in the $i$th $s$-tuple also appears in the $j$th $s$-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 $i$th of the $r$ $s$-tuples survives the restriction is maximised when each variable among the $s$ describes a different pigeon (by Lemma 20) and is therefore bound above by

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

whereupon

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

which is $\leq 1/e^{\frac{n}{4^{\xi(s)+1}s!\cdot 2^t u^{\xi(s)}}}$. 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 $\mathsf{Res}(s)$ refutation of some $\mathsf{Bin\text{-}PHP}^m_n\restriction_\sigma$, where $\sigma$ 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 $\mathsf{Res}(s)$ refutation is of size $< e^{\frac{n}{4^{\xi(s)+1}s!\cdot 2^t u^{\xi(s)}}}$ then, from Lemma 1, it would give a $\mathsf{Res}(s-1)$ refutation of size

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

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

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

**Proof.** First, let us claim that $\mathsf{PHP}(\sqrt{2}\log^{\frac{1}{2+\lambda}} n, 0)$ holds (and this would hold also at $\lambda = 0$). Applying Theorem 21 gives $\ell$ such that $\frac{\ell(\ell+1)}{2} < \log n$. Noting $\frac{\ell^2}{2} < \frac{\ell(\ell+1)}{2}$, 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 $\xi(s) = \Theta(s^2)$. It follows, since $\lambda > 0$, that each of $4^{\xi(s)+1}$, $s!$ and $\log^{\xi(s)} n$ is $o(n^\mu)$. The result follows. ◀

## 4.1 The treelike case

Concerning the Pigeonhole Principle, we can prove that the relationship between PHP and Bin-PHP 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 23.** *The treelike Resolution complexity of $\mathsf{Bin\text{-}PHP}^m_n$ is $2^{\Theta(n)}$.*

**Proof.** For the lower bound, one can follow the proof of Lemma 16 with $t = 0$ and finds $n$ free choices on each branch of the tree. Following the method of Riis [34], 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 $h^0_1$. At least one of $|T_1| > h^1_1$ or $|F_1| > h^0_1$ 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 $\log n$ iteration since we always choose a set of maximal size. The depth of our tree is bound above by $n + \frac{n}{2} + \frac{n}{4} + \cdots < 2n$ and the result follows. ◀

# 5 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 [34] we consider principles which are expressible as first order formulae with no finite model in $\Pi_2$-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 the *Ordering*

*Principle*, which states that a finite partial order has a maximal element is one of such principle. Its negation can be expressed in $\Pi_2$-form 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 $\mathsf{OP_n}$ using a *unary encoding* of the witness, as shown below. In Definition 25 we explain how to generate a binary encoding $\mathsf{Bin\text{-}C_n}$ from any combinatorial principle $\mathsf{C_n}$ expressible as a first order formulae in $\Pi_2$-form with no finite models and whose unary encoding we denote by $\mathsf{Un\text{-}C_n}$. For example $\mathsf{Bin\text{-}OP_n}$ would be the conjunction of the clauses below.

$\mathsf{OP_n}$ : *Unary encoding*

$$\begin{array}{ll} \overline{v}_{x,x} & x \in [n] \\ \overline{v}_{x,y} \vee \overline{v}_{y,z} \vee v_{x,z} & x,y,z \in [n] \\ \bigvee_{i \in [n]} v_{x,i} & x \in [n] \end{array}$$

$\mathsf{Bin\text{-}OP_n}$ : *Binary encoding*

$$\begin{array}{ll} \overline{\nu}_{x,x} & x \in [n] \\ \overline{\nu}_{x,y} \vee \overline{\nu}_{y,z} \vee \nu_{x,z} & x,y,z \in [n] \\ \bigvee_{i \in [\log n]} \omega_{x,i}^{1-a_i} \vee \nu_{x,a} & x,a \in [n] \\ a_1 \ldots a_{\log n} & \text{binary representation of } a \end{array}$$

$$\omega_{x,j}^{a_j} = \begin{cases} \omega_{x,j} & a_j = 1 \\ \overline{\omega}_{x,j} & a_j = 0 \end{cases}$$

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 [34], for $m = n+1$, the negation of its relational form can be expressed as a $\Pi_2$-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 are:

$\mathsf{PHP}$ : *Unary encoding*

$$\begin{array}{ll} \bigvee_{j=1}^{n} v_{i,j} & i \in [m] \\ \overline{v}_{i,j} \vee \overline{v}_{i',j} & i, \neq i' \in [m], j \in [n] \end{array}$$

$\mathsf{Bin\text{-}PHP}$ : *Binary encoding*

$$\bigvee_{j=1}^{\log n} \overline{\omega}_{i,j} \vee \bigvee_{j=1}^{\log n} \overline{\omega}_{i',j} \quad i \neq i' \in [m]$$

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 account 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.

## 5.1 Binary encodings of principles involving total comparison

We will now argue that the proof complexity in Resolution of principles involving total comparison will not increase significantly (by more than a polynomial factor) when shifting from the unary encoding to the binary encoding. *Total comparison* is here indicated by the axioms $v_{i,j} \oplus v_{j,i}$, where $\oplus$ indicates XOR, for each $i \neq j$. It follows that it does not make sense to consider the binary encoding of such principles in the search for strong lower bounds. Examples of natural principles involving total comparison include the totally ordered variant of the Ordering Principle (known to be polynomially refutable in Resolution [14]) as well as all of its unary relativisations (which can be exponentially hard for any $\mathrm{Res}(s)$ [17]).

Let $\mathsf{TC\text{-}Prin}$ be some $\Pi_2$ first-order principle involving relations of arity no more than 2. Let $n \in \mathbb{N}$ and discover $\mathsf{TC\text{-}Prin}(n)$ with variables $v_{i,j}$, for $i,j \in [n]$, of arity 2, including

axioms of total comparison: $v_{i,j} \oplus v_{j,i}$, for each $i \neq j$. There may additionally be unary variables, of the form $u_i$, for $i \in [n]$, but no further variables of other arity. Let Un-TC-Prin$(n)$ have axioms $v_{i,1} \vee \ldots \vee v_{i,n}$, for each $i \in [n]$ (for the Ordering Principle this would most naturally correspond to the variant stating a finite total order has a maximal element). To make our translation to the binary encoding, we tacitly assume $n$ is a power or 2. When this is not the case, we need clauses forbidding certain evaluations, and we defer this treatment to Section 5.2. Let Bin-TC-Prin$(n)$ have corresponding variables $\omega_{i,\ell}$ for $i \in [n], \ell \in [\log n]$, where $v_{i,j}$ from the unary encoding semantically corresponds to the conjunction $(\omega_{i,1}^{a_1} \wedge \ldots \wedge \omega_{i,\log n}^{a_{\log n}})$, where

$$\omega_{i,p}^{a_p} = \begin{cases} \omega_{i,p} & \text{if } a_p = 1 \\ \overline{\omega}_{i,p} & \text{if } a_p = 0 \end{cases}$$

with $a_1 \cdots a_{\log n}$ being the binary representation of $j$. The unary variables stay as they are. From this, the axioms of Bin-TC-Prin$(n)$, including total comparison, can be canonically calculated from the corresponding axioms of Un-TC-Prin$(n)$ as explained in Section 5.2 in Defintion 25. Note that the large disjunctive clauses of Un-TC-Prin$(n)$, that encode the existence of the witness, disappear completely in Bin-TC-Prin$(n)$.

▶ **Lemma 24.** *Suppose there is a Resolution refutation of* Un-TC-Prin$(n)$ *of size* $S(n)$*. Then there is a Resolution refutation of* Bin-TC-Prin$(n)$ *of size at most* $n^2 \cdot S(n)$*.*

**Proof.** Take a decision DAG $\pi$ for Un-TC-Prin$(n)$ and consider the point at which some variable $v_{i,j}$ is questioned. Each node in $\pi$ will be expanded to a small tree in $\pi'$, which will be a decision DAG for Bin-TC-Prin$(n)$. The question "$v_{i,j}$?" in $\pi$ will become a sequence of $2 \log n$ questions on variables $\omega_{i,1}, \ldots, \omega_{i,\log n}, \omega_{j,1}, \ldots, \omega_{j,\log n}$, giving rise to a small tree of size $2^{2 \log n} = n^2$ questions in $\pi'$. Owing to total comparison, many of the branches of this mini-tree must end in contradiction. Indeed, many of their leaves would imply the impossible $\neg v_{i,j} \wedge \neg v_{j,i}$, while precisely one would imply the impossible $v_{i,j} \wedge v_{j,i}$ (see Figure 1 for an example). Those that don't will always have a sub-branch labelled by $(\omega_{i,1}^{a_1} \wedge \ldots \wedge \omega_{i,\log n}^{a_{\log n}})$, where

$$\omega_{i,p}^{a_p} = \begin{cases} \omega_{i,p} & \text{if } a_p = 1 \\ \overline{\omega}_{i,p} & \text{if } a_p = 0 \end{cases}$$

with $a_1 \cdots a_{\log n}$ being the binary representation of $j$; **or** $(\omega_{j,1}^{b_1} \wedge \ldots \wedge \omega_{j,\log n}^{b_{\log n}})$, where

$$\omega_{j,p}^{b_p} = \begin{cases} \omega_{j,p} & \text{if } b_p = 1 \\ \overline{\omega}_{j,p} & \text{if } b_p = 0 \end{cases}$$

with $b_1 \cdots b_{\log n}$ being the binary representation of $i$. By forgetting information along these branches and unifying branches with the same labels of their sub-branches, we are left with precisely these two outcomes, corresponding to "$v_{i,j}$" and "$\neg v_{i,j}$", which is "$v_{j,i}$". Indeed, this is the crux, $\neg v_{i,j}$ being equivalent to $v_{j,i}$, and thus being expressible as some conjunction of variables $\omega_{j,p}^{b_p}$. Thus, $\pi$ gives rise to $\pi'$ of size $n^2 \cdot S(n)$ and the result follows. ◀

## 5.2 Binary versus unary encodings in general

Let $\mathsf{C_n}$ be some combinatorial principle expressible as a first-order $\Pi_2$-formula $F$ of the form $\forall \vec{x} \exists \vec{w} \varphi(\vec{x}, \vec{w})$ where $\varphi(\vec{x}, \vec{w})$ is a quantifier-free formula built on a family of relations $\vec{R}$. Following Riis [34] we restrict to the class of such formulae having no finite model.

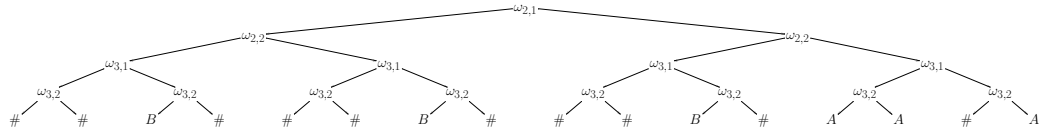


**Figure 1** Example converting the question $v_{2,3}$? from a Resolution refutation of Un-TC-Prin($n$) to a small tree in a refutation of Bin-TC-Prin($n$). The variables $\omega_{2,1}, \omega_{2,2}, \omega_{3,1}, \omega_{3,2}$ are questioned in order. The left-hand and right-hand branches correspond to false and true, respectively. Note that 2 and 3 are 10 and 11 in binary, respectively. Thus, $v_{2,3}$ is equivalent to $\omega_{2,1} \wedge \omega_{2,2}$ (labelled $A$ at the leaves) and $v_{3,2}$ is equivalent to $\omega_{3,1} \wedge \overline{\omega}_{3,2}$ (labelled $B$ at the leaves). The remaining leaves contradict the total comparison clauses (including one that would be labelled both $A$ and $B$).

Let Un-$\mathsf{C_n}$ be the standard unary (see Riis in [34]) CNF propositional encoding of $F$. For each set of first-order variables $\vec{a} := \{x_1, \ldots, x_k\}$ of (first order) variables, we consider the propositional variables $v_{x_{i_1}, x_{i_2}, \ldots, x_{i_k}}$ (which we abbreviate as $v_{\vec{a}}$) whose semantics are to capture at once the value of variables in $\vec{a}$ if they appear in some relation in $\varphi$. For easiness of description we restrict to the case where $F$ is of the form $\forall \vec{x} \exists w \varphi(\vec{x}, w)$, i.e. $\vec{w}$ is a single variable $w$. Hence the propositional variables of Un-$\mathsf{C_n}$ are of the type $v_{\vec{a}}$ for $\vec{a} \subseteq \vec{x}$ (type 1 variables) and/or of the type $v_{\vec{x}w}$ for $w \in \vec{w}$ (type 2 variables) and which we denote by simply $v_w$, since each existential variable in $F$ depends always on all universal variables. Notice that we consider the case of $F = \forall \vec{x} \exists w \varphi(\vec{x}, w)$, since the generalisation to higher arity is clear as each witness $w \in \vec{w}$ may be treated individually.

▶ **Definition 25** (Canonical form of Bin-$\mathsf{C_n}$). *Let* $\mathsf{C_n}$ *be a combinatorial principle expressible as a first-order formula* $\forall \vec{x} \exists w \varphi(\vec{x}, w)$ *with no finite models. Let* Un-$\mathsf{C_n}$ *be its unary propositional encoding. Let* $2^{r-1} < n \leq 2^r \in \mathbb{N}$ $(r = \lceil \log n \rceil)$. *The binary encoding* Bin-$\mathsf{C_n}$ *of* $C$ *is defined as follows:*

*The* ***variables*** *of* Bin-$\mathsf{C_n}$ *are defined from variables of* Un-$\mathsf{C_n}$ *as follows:*

1. *For each variable of type 1* $v_{\vec{a}}$, *for* $\vec{a} \subseteq \vec{x}$, *we use a variable* $\nu_{\vec{x}}$, *for* $\vec{a} \subseteq \vec{x}$, *and*
2. *For each variable of type 2* $v_w$, *we have* $r$ *variables* $\omega_1, \ldots \omega_r$, *where we use the convention that if* $z_1 \ldots z_r$ *is the binary representation of* $w$, *then*

$$\omega_j^{z_j} = \begin{cases} \omega_j & z_j = 1 \\ \overline{\omega}_j & z_j = 0 \end{cases}$$

*so that* $v_w$ *can be represented using binary variables by the clause* $(\omega_1^{1-z_1} \vee \ldots \vee \omega_r^{1-z_r})$

*The* ***clauses*** *of* Bin-$\mathsf{C_n}$ *are defined form the clauses of* Un-$\mathsf{C_n}$ *as follows:*

1. *If* $C \in$ Un-$\mathsf{C_n}$ *contains only variables of type 1,* $v_{\vec{b}_1}, \ldots, v_{\vec{b}_k}$, *hence* $C$ *is mapped as follows*

$$C := \bigvee_{j=1}^{k_1} v_{\vec{b}_j} \vee \bigvee_{j=1}^{k_2} \overline{v}_{\vec{c}_j} \quad \mapsto \quad \bigvee_{j=1}^{k_1} \nu_{\vec{b}_j} \vee \bigvee_{j=1}^{k_2} \overline{\nu}_{\vec{c}_j}$$

2. *If* $C \in$ Un-$\mathsf{C_n}$ *contains type 1 and type 2 variables, it is mapped as follows:*

$$C := v_w \vee \bigvee_{j=1}^{k_1} v_{\vec{c}_j} \vee \bigvee_{l=1}^{k_2} \overline{v}_{\vec{d}_j} \quad \mapsto \quad \left(\bigvee_{i \in [r]} \omega_i^{1-z_i}\right) \vee \bigvee_{j=1}^{k_1} \nu_{\vec{c}_j} \vee \bigvee_{l=1}^{k_2} \overline{\nu}_{\vec{d}_j}$$
$$C := \overline{v}_w \vee \bigvee_{j=1}^{k_1} v_{\vec{c}_j} \vee \bigvee_{l=1}^{k_2} \overline{v}_{\vec{d}_j} \quad \mapsto \quad \left(\bigvee_{i \in [r]} \omega_i^{z_i}\right) \vee \bigvee_{j=1}^{k_1} \nu_{\vec{c}_j} \vee \bigvee_{l=1}^{k_2} \overline{\nu}_{\vec{d}_j}$$

*where* $\vec{c}_j, \vec{d}_l \subseteq \vec{x}$ *and where* $z_1, \ldots, z_r$ *is the binary representation of* $w$.

3. *If* $n \neq 2^r$, *then, for each* $n < a \leq 2^r$ *we need clauses*

$$\omega_1^{1-a_1} \vee \ldots \vee \omega_r^{1-a_r}$$

*where* $a_1, \ldots, a_r$ *is the binary representation of* $a$.

Getting short proofs for the binary version Bin-$\mathsf{C_n}$ in Res($\log n$) form short Res(1) proofs of the unary version Un-$\mathsf{C_n}$ is possible also in the general case.

▶ **Lemma 26.** *Let* $\mathsf{C_n}$ *be a combinatorial principle expressible as a first-order formula* $\forall\vec{x}\exists\vec{w}\varphi(\vec{x},\vec{w})$ *with no finite models. Let* Un-$\mathsf{C_n}$ *and* Bin-$\mathsf{C_n}$ *be respectively the unary and binary propositional encoding. Let* $n \in \mathbb{N}$*. If there is a size* $S$ *refutation for* Un-$\mathsf{C_n}$ *in* Res(1)*, then there is a size* $S$ *refutation for* Bin-$\mathsf{C_n}$ *in* Res($\log n$)

**Proof Sketch.** Where the decision DAG for Un-$\mathsf{C_n}$ questions some variable $v_{\vec{a},b}$, the decision branching $\log n$-program questions instead $(\omega_{\vec{a},1}^{1-z_1} \vee \ldots \vee \omega_{\vec{a},\log n}^{1-z_{\log n}})$ 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 Bin-$\mathsf{C_n}$, and the result follows. ◀

As one can easily notice reading Subsection 1.3, the binary version Bin-PHP of the Pigeonhole Principle we displayed there, is different from the one we would get applying the canonical transformation of Definition 5.2. However, we can easily and efficiently move between these versions in Resolution (we leave the proof to the reader below), and the version we have chosen is easier to handle, having fewer variables.

▶ **Lemma 27.** *The two versions of the binary Pigeonhole Principle (*Bin-PHP *and the one arising from Definition 5.2 to* PHP*) are linearly equivalent in Resolution.*

## 5.3 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 28.** *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 $\pi'$ for Bin-C($n$), where w.l.o.g. $n$ is even, and consider the point at which some variable $\nu'_{i,j}$ is questioned. Each node in $\pi'$ will be expanded to a small tree in $\pi$, which will be a decision DAG for Un-Fun-C($n$). The question "$\nu'_{i,j}$?" in $\pi$ 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. Suppose $v_{i,k}$ is true. If the $j$th 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 $j$th bit is 0. All of these must be false. Likewise, if the $j$th 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 $j$th 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, $\pi'$ gives rise to $\pi$ of size $n^2 \cdot S(n)$ and the result follows. ◀

## 5.4 The Ordering Principle in binary

Recall the Ordering Principle specified in $\Pi_2$ first-order logic

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

with propositional translation to the binary encoding of witnesses, Bin-OP$_n$, as follows.

$$\begin{array}{ll} \overline{\nu}_{x,x} & x \in [n] \\ \overline{\nu}_{x,y} \vee \overline{\nu}_{y,z} \vee \nu_{x,z} & x,y,z \in [n] \\ \bigvee_{i \in [\log n]} \omega_{x,i}^{1-a_i} \vee \nu_{x,a} & x,a \in [n] \end{array}$$

where

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

and $a_1 \dots a_{\log n}$ is the binary representation of $a$.

▶ **Lemma 29.** Bin-OP$_n$ *has refutations in Resolution of polynomial size.*

**Proof.** We follow the well-known proof for the unary version of the Ordering Principle, from [36]. Consider the domain to be $[n] = \{1, \dots, n\}$. At the $i$th stage of the decision DAG we will find a maximal element, ordered by $R$, among $[i] = \{1, \dots, i\}$. That is, we will have a record of the *special* form

$$\overline{\nu}_{j,1} \wedge \dots \wedge \overline{\nu}_{j,j-1} \wedge \overline{\nu}_{j,j+1} \wedge \dots \wedge \overline{\nu}_{j,i}$$

for some $j \in [i]$. The base case $i = 1$ is trivial. Let us explain the inductive step. From the displayed record above we ask the question $\nu_{j,i+1}$? If $\nu_{j,i+1}$ is true, then ask the sequence of questions $\nu_{i+1,1}, \dots, \nu_{i+1,i}$, all of which must be false by transitivity. Now, by forgetting information, we uncover a new record of the special form. Suppose now $\nu_{j,i+1}$ is false. Then we equally have a new 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 records and navigating between special records involves a path of length $n$, so we have a DAG of size $n^3$. Finally, at $i = n$, we have a record of the form

$$\overline{\nu}_{j,1} \wedge \dots \wedge \overline{\nu}_{j,j-1} \wedge \overline{\nu}_{j,j+1} \wedge \dots \wedge \overline{\nu}_{j,n}.$$

Now we expand a tree questioning the sequence $w_{j,1}, \dots, w_{j,\log n}$, 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 $n^3 + n^2$. ◀

▶ **Theorem 30.** Bin-OP$_n$ *has poly size resolution refutations in* Res(1).

## 6 Final remarks

Various questions are left unanswered in 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 $s = \Theta(\log \log n)$ for the former, and $s = o(\log^{1/2} n)$ 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.

Similarly, one might ask whether converses to our lemmas might hold. For example, to Lemmas 24 and 26. In these cases, we do not know about the converses. The converse of Lemma 28(even for $n^2$ replaced by some polynomial) is false. For example, consider the very weak Pigeonhole Principle of [15].

## References


1 Michael Alekhnovich. Lower Bounds for k-DNF Resolution on Random 3-CNFs. *Computational Complexity*, 20(4):597–614, 2011. doi:10.1007/s00037-011-0026-0.

2 Razborov Alexander A. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, 181(41):415–472, 2015.

3 Albert Atserias. Improved bounds on the Weak Pigeonhole Principle and infinitely many primes from weaker axioms. *Theor. Comput. Sci.*, 295:27–39, 2003. doi:10.1016/S0304-3975(02)00394-8.

4 Albert Atserias, Ilario Bonacina, Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Alexander A. Razborov. Clique is hard on average for regular resolution. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 866–877. ACM, 2018. doi:10.1145/3188745.3188856.

5 Albert Atserias, Maria Luisa Bonet, and Juan Luis Esteban. Lower Bounds for the Weak Pigeonhole Principle and Random Formulas beyond Resolution. *Inf. Comput.*, 176(2):136–152, 2002. doi:10.1006/inco.2002.3114.

6 Paul Beame, Russell Impagliazzo, and Ashish 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*, pages 52–68. IEEE Computer Society, 2001. doi:10.1109/CCC.2001.933872.

7 Paul Beame and Toniann 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*, pages 274–282. IEEE Computer Society, 1996. doi:10.1109/SFCS.1996.548486.

8 Eli Ben-sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. In *Journal of the ACM*, pages 517–526, 1999.

9 Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. A lower bound for the pigeonhole principle in tree-like Resolution by asymmetric Prover-Delayer games. *Inf. Process. Lett.*, 110(23):1074–1077, 2010. doi:10.1016/j.ipl.2010.09.007.

10 Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. Parameterized Complexity of DPLL Search Procedures. *ACM Trans. Comput. Logic*, 14(3):20:1–20:21, August 2013. doi:10.1145/2499937.2499941.

11 Olaf Beyersdorff, Nicola Galesi, Massimo Lauria, and Alexander A. Razborov. Parameterized Bounded-Depth Frege Is not Optimal. *TOCT*, 4(3):7:1–7:16, 2012. doi:10.1145/2355580.2355582.

12 Ilario Bonacina and Nicola Galesi. A Framework for Space Complexity in Algebraic Proof Systems. *J. ACM*, 62(3):23:1–23:20, 2015. doi:10.1145/2699438.

13 Ilario Bonacina, Nicola Galesi, and Neil Thapen. Total Space in Resolution. *SIAM J. Comput.*, 45(5):1894–1909, 2016. doi:10.1137/15M1023269.

14 Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, 2001. doi:10.1007/s000370100000.

15 Samuel R. Buss and Toniann 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*, pages 149–156, 1997. doi:10.1007/BFb0028012.

16 Stefan S. Dantchev and Søren 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*, pages 69–75, 2001. doi:10.1109/CCC.2001.933873.

17 Stefan S. Dantchev and Søren Riis. On Relativisation and Complexity Gap. In Matthias Baaz and Johann A. Makowsky, editors, *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*, volume 2803 of *Lecture Notes in Computer Science*, pages 142–154. Springer, 2003. doi:10.1007/978-3-540-45220-1_14.

**18** Juan Luis Esteban, Nicola Galesi, and Jochen Messner. On the complexity of resolution with bounded conjunctions. *Theor. Comput. Sci.*, 321(2-3):347–370, 2004. doi:10.1016/j.tcs.2004.04.004.

**19** Yuval Filmus, Massimo Lauria, Jakob Nordström, Noga Ron-Zewi, and Neil Thapen. Space Complexity in Polynomial Calculus. *SIAM J. Comput.*, 44(4):1119–1153, 2015. doi:10.1137/120895950.

**20** Nicola Galesi and Massimo Lauria. Optimality of size-degree tradeoffs for polynomial calculus. *ACM Trans. Comput. Log.*, 12(1):4:1–4:22, 2010. doi:10.1145/1838552.1838556.

**21** Armin Haken. The Intractability of Resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

**22** Pavel Hrubes and Pavel Pudlák. Random Formulas, Monotone Circuits, and Interpolation. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 121–131. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.20.

**23** Jan Krajíček. *Bounded arithmetic, propositional logic and complexity theory*. Cambridge University Press, 1995.

**24** Balakrishnan Krishnamurthy. Short Proofs for Tricky Formulas. *Acta Inf.*, 22(3):253–275, 1985. doi:10.1007/BF00265682.

**25** Oliver 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)*, 41, 1999. URL: http://eccc.hpi-web.de/eccc-reports/1999/TR99-041/index.html.

**26** 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.

**27** Massimo Lauria, Pavel Pudlák, Vojtech Rödl, and Neil Thapen. The complexity of proving that a graph is Ramsey. *Combinatorica*, 37(2):253–268, 2017. doi:10.1007/s00493-015-3193-9.

**28** Alexis Maciel, Toniann Pitassi, and Alan R. Woods. A New Proof of the Weak Pigeonhole Principle. *J. Comput. Syst. Sci.*, 64(4):843–872, 2002. doi:10.1006/jcss.2002.1830.

**29** Justyna Petke. *Bridging Constraint Satisfaction and Boolean Satisfiability*. Artificial Intelligence: Foundations, Theory, and Algorithms. Springer, 2015. doi:10.1007/978-3-319-21810-6.

**30** P. Pudlák. Proofs as games. *American Mathematical Monthly*, pages 541–550, June-July 2000.

**31** Ran Raz. Resolution lower bounds for the weak pigeonhole principle. *J. ACM*, 51(2):115–138, 2004. doi:10.1145/972639.972640.

**32** Alexander A. Razborov. Proof Complexity of Pigeonhole Principles. In Werner Kuich, Grzegorz Rozenberg, and Arto Salomaa, editors, *Developments in Language Theory*, pages 100–116, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

**33** Alexander A. Razborov. Resolution lower bounds for the weak functional pigeonhole principle. *Theor. Comput. Sci.*, 1(303):233–243, 2003. doi:10.1016/S0304-3975(02)00453-X.

**34** Søren Riis. A complexity gap for tree resolution. *Computational Complexity*, 10(3):179–209, 2001. URL: http://link.springer.de/link/service/journals/00037/bibs/1010003/10100179.htm.

**35** Nathan Segerlind, Samuel R. Buss, and Russell Impagliazzo. A Switching Lemma for Small Restrictions and Lower Bounds for k-DNF Resolution. *SIAM J. Comput.*, 33(5):1171–1200, 2004. doi:10.1137/S0097539703428555.

**36** Gunnar Stålmarck. Short Resolution Proofs for a Sequence of Tricky Formulas. *Acta Inf.*, 33(3):277–280, 1996. doi:10.1007/s002360050044.

**37** Toby Walsh. SAT v CSP. In *Principles and Practice of Constraint Programming - CP 2000, 6th International Conference, Singapore, September 18-21, 2000, Proceedings*, pages 441–456, 2000. doi:10.1007/3-540-45349-0_32.