Electronic Colloquium on Computational Complexity, Report No. 165 (2018)

Resolution and the binary encoding of combinatorial principles

Stefan Dantchev[1], Nicola Galesi[2], and Barnaby Martin[1]

[1]Department of Computer Science, University of Durham
[2]Dipartimento di Informatica, Sapienza Università Roma

September 19, 2018

Abstract

We investigate the size complexity of proofs in $\mathsf{Res}(s)$ – an extension of Resolution working on s-DNFs instead of clauses – for families of contradictions given in the *unusual binary* encoding. A motivation of our work is size lower bounds of refutations in Resolution for families of contradictions in the *usual unary* encoding. Our main interest is the k-Clique Principle, whose Resolution complexity is still unknown. The approach is justified by the observation that for a large class of combinatorial principles (those expressible as Π_2 first-order formulae) short $\mathsf{Res}(\log n)$ refutations for the binary encoding are reducible to short Resolution refutations of the unary encoding.

Our main result is a $n^{\Omega(k)}$ lower bound for the size of refutations of the binary k-Clique Principle in $\mathsf{Res}(\lfloor \frac{1}{2} \log \log n \rfloor)$. This improves the result of Lauria, Pudlák et al. [24] who proved the lower bound for Resolution, that is $\mathsf{Res}(1)$. A lower bound in $\mathsf{Res}(\log n)$ for the binary k-Clique Principle would prove a lower bound in Resolution for its unary version. Resolution lower bounds for the (unary) k-Clique Principle are known only when refutations are either treelike [10] or read-once [4] (regular Resolution).

To contrast the proof complexity between the unary and binary encodings of combinatorial principles, we consider the binary (weak) Pigeonhole principle $\mathsf{Bin\text{-}PHP}^m_n$ for $m > n$. Our second lower bound proves that in $\mathsf{Res}(s)$ for $s \leq \log^{\frac{1}{2-\epsilon}}(n)$, the shortest proofs of the $\mathsf{Bin\text{-}PHP}^m_n$, requires size $2^{n^{1-\delta}}$, for any $\delta > 0$.

By a result of Buss and Pitassi [15] we know that for the (unary, weak) Pigeonhole principle PHP^m_n, exponential lower bounds (in the size of PHP^m_n) are not possible in Resolution when $m \geq 2^{\sqrt{n \log n}}$ since there is an upper bound of $2^{O(\sqrt{n \log n})}$. Our lower bound for $\mathsf{Bin\text{-}PHP}^m_n$, together with the fact short $\mathsf{Res}(1)$ refutations for PHP^m_n can be translated into short $\mathsf{Res}(\log n)$ proofs for $\mathsf{Bin\text{-}PHP}^m_n$, shows a form of tightness of the upper bound of [15]. Furthermore we prove that $\mathsf{Bin\text{-}PHP}^m_n$ can be refuted in size $2^{\Theta(n)}$ in treelike $\mathsf{Res}(1)$, contrasting with the unary case, where PHP^m_n requires treelike $\mathsf{Res}(1)$ refutations of size $2^{\Omega(n \log n)}$ [9, 16].

In order to compare the complexity of refuting binary encodings in Resolution with respect to their unary version, we study under what conditions the complexity of refutations in Resolution will not increase significantly (more than a polynomial factor) when shifting between the unary encoding and the binary encoding. We show that this is true, from unary to binary, for propositional encodings of principles expressible as a Π_2-formula and involving *total variable comparisons*. We then show that this is true, from binary to unary, when one considers the *functional unary encoding*. In particular, we derive a polynomial upper bound in $\mathsf{Res}(1)$ for the binary version $\mathsf{Bin\text{-}rLOP_n}$ of a variant of the Linear Ordering principle, $\mathsf{rLOP_n}$, which exponentially separates read-once Resolution from Resolution (see [2]).

Finally we prove that the binary encoding of the general Ordering principle $\mathsf{Bin\text{-}OP_n}$ – with no total ordering constraints – is polynomially provable in Resolution. These last results can be interpreted as addressing the property that shifting to the binary encoding is preserving the proof hardness of the corresponding unary encodings when working in Resolution.

ISSN 1433-8092

1 Introduction

Various fundamental combinatorial principles used in Proof Complexity may be given in first-order logic as sentences φ with no finite models. Riis discusses in [30] how to generate from φ a family of CNFs, the nth of which encodes that φ 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, 24, 21], 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 gives a witness, whereas in the unary encoding a solitary true literal tells us which is the witness[1]. 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, 24].

The central thrust of this work is to contrast the proof complexity (size) between the unary and binary encodings of natural combinatorial principles. The main 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 Krajîĉek in [22]. Our approach is justified by observing that (see Lemma 14), for a family of contradictions encoding a principle which is expressible as Π_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.

Our main 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 Formulas, $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$; (2) the (weak) Pigeonhole Principle PHP^{m}_{n}; and (3) the (Linear) Ordering Principle, (L)OP_{n}. The *k-Clique Formulas* introduced in [10, 11, 6] are formulas stating that a given graph G does have a k-clique and are therefore unsatisfiable when G does not contain a k-clique. The Pigeonhole principle states that a total mapping $f : [m] \rightarrow [n]$ has necessarily a collision when $m > n$. Its propositional formulation in the negation, PHP^{m}_{n} is well-studied in proof complexity (see among others: [20, 31, 16, 27, 29, 28, 8, 15, 9, 7, 5, 3, 25]). The $\mathsf{LOP_n}$ formulas encodes 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 [23, 32, 14].

1.1 Contributions

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

[1] see Subsection 1.1.2 in the Introduction for examples and a more formal statement.

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 [24], 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 1) is a $n^{\Omega(k)}$ lower bound for the size of refutations of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ in $\mathsf{Res}(\frac{1}{2}\log\log n)$, 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 $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$.

1.1.1 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 $s \leq \log^{\frac{1}{2-\epsilon}}(n)$, the shortest proofs of the $\mathsf{Bin\text{-}PHP}^m_n$, require size $2^{n^{1-\delta}}$, for any $\delta > 0$ (Theorem 4). 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 2) 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 [31]. Since we are not using any (even weak) form of Switching Lemma (as for instance in [31, 1]), we consider how tight is our lower bound in $\mathsf{Res}(s)$. We prove that $\mathsf{Bin\text{-}PHP}^m_n$ (Theorem 5) 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, PHP^m_n, which instead requires treelike $\mathsf{Res}(1)$ refutations of size $2^{\Omega(n\log n)}$, as proved in [9, 16].

As for the k-Clique principle, also for the Pigeonhole 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 PHP^m_n (Lemma 4). Hence another observation raising from our lower bound concerns the result of Buss and Pitassi in [15], who proved a quasipolynomial upper bounds (in the number of variables of PHP^m_n) for the size of refuting PHP^m_n when $m \geq 2^{\sqrt{n\log n}}$. Indeed, they give the subexponential-in-n upper bound of $2^{O(\sqrt{n\log n})}$. Hence no exponential-in-n lower bound is possible in Resolution when $m \geq 2^{\sqrt{n\log n}}$. Since we prove that $\mathsf{Bin\text{-}PHP}^m_n$ requires $2^{n^{1-\delta}}$ size in $\mathsf{Res}(s)$ for any $m > n$, then Lemma 4 is indicating that Buss and Pitass's result in [15] is essentially tight and cannot be be proved for the binary version of the Pigeonhole principle.

1.1.2 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 [30] we consider principles which are expressible as first order formulas with no finite model in Π_2-form, i.e. as $\forall\vec{x}\exists\vec{w}\varphi(\vec{x},\vec{w})$ where $\varphi(\vec{x},\vec{y})$ is a formula built on a family of relations $\vec{R}$. For example 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 Π_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 OP_n using a *unary encoding* of the witness, as shown below. In Definition 4 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 formulas in Π_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*

$$\begin{array}{ll} \overline{\nu}_{x,x} & x \in [n] \\ \overline{\nu}_{x,y} \vee \overline{\nu}_{y,z} \vee \nu_{x,z} & x,y,z \in [n] \\ \bigvee_{i\in[\log n]} \omega_{x,i}^{1-a_i} \vee \nu_{x,a} & x,a \in [n] \\ a_1 \dots a_{\log n} \text{ binary representation of } a & \\ \omega_{x,j}^{a_j} = \begin{cases} \omega_{x,j} & a_j = 1 \\ \overline{\omega}_{x,j} & a_j = 0 \end{cases} & \end{array}$$

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 [30] the negation of its relational form can be expressed as a Π_2-formula as

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

and its usual unary and binary propositional encoding are:

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. In Section 6 we show that binary encodings are most interesting to study for Π_2 combinatorial principles *all of whose witnesses are of a different type from the variables they are witnesses for.*

In Section 6 we observe that Lemma 2 and 8 work also for the general case of $\mathsf{Un\text{-}C_n}$ and $\mathsf{Bin\text{-}C_n}$ (Lemma 14). We also prove in Lemma 4 that the usual binary encoding Bin-PHP of the PHP ([19, 12]) is provably equivalent in Resolution to the version of the binary version Pigeonhole principle defined from our translation to binary of Definition 4. We finally 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 $\mathsf{Res}(1)$.

1.1.3 Total comparisons and Linear Ordering principles

$\mathsf{LOP_n}$ formulae took on a certain importance in Resolution. In their more general form they were used in [14] to prove the optimality of the size-width tradeoffs for Resolution (see [8]). More importantly for this work, a modification of the $\mathsf{LOP_n}$ formulas ($\mathsf{rLOP_n}$) were used in [2] to exhibit a family of formulas exponentially separating proof size in read-once Resolution from Resolution.

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 11 we prove that this is true for the negation of principles expressible as first order formula in Π_2-form involving *total variable comparisons*. Hence in particular (see Corollary 3) the binary version of the Linear Ordering principle $\mathsf{Bin\text{-}LOP_n}$ and its modification $\mathsf{Bin\text{-}rLOP_n}$ which separates read-once Resolution from Resolution (see [2]) are polynomially provable in Resolution. It is worthy to notice that $\mathsf{Bin\text{-}rLOP_n}$ is polynomially provable in $\mathsf{Res}(\frac{1}{2}\log\log n)$, where we prove a lower bound for $\mathsf{Bin\text{-}Clique^n_k}(G)$.

Finally, we also prove that the binary encoding of the general Linear Ordering $\mathsf{Bin\text{-}OP_n}$ principle, where antisymmetry – which entails total comparisons – is not encoded, is also polynomially provable in Resolution. Ordering Principles are typically used to provide hierarchy separations (see for instance [31, 17]) inside Resolution-based proof systems. Hence, loosely speaking, they mark the maximal border of what is still provable efficiently in a given proof system. For this reason the upper bounds explained in this subsection for the binary version of the ordering principles should be interpreted broadly speaking, as saying that shifting to the binary encodings is not destroying the hardness of a unary principle when working in Resolution and hence binary encodings of combinatorial principles are still meaningful benchmarks to prove lower bounds for.

1.1.4 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 5.2, 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 5), the unary (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.

1.2 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 [31, 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) Pigeohole 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 Π_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.

We can now describe the propositional refutation system $\mathsf{Res}\,(s)$ ([22]). 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 π of $\mathcal{C}$ the width of π, $w(\pi)$ is the maximal width of a clause in π. The width refuting $\mathcal{C}$ in $\mathsf{Res}(1)$, $w(\vdash \mathcal{C})$, 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 for 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 $\mathcal{F}(x_1 \ldots, x_n)$ be a boolean s-DNF (resp. s-CNF) defined over variables $X = \{x_1, \ldots, x_n\}$. A *partial assignment* ρ 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 $\mathsf{Res}\,(s)$ refutation of a given set of s-clauses Σ 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. 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 [26]). 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.

$$\begin{tikzcd}
& \mathcal{C} & \\
& ?\bigvee_{j\in J} l_j \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\
\mathcal{C}\land\bigvee_{j\in J} l_j & & \mathcal{C}\land\bigwedge_{j\in J}\neg l_j
\end{tikzcd} \tag{1}$$

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

$$\begin{tikzcd}
& \mathcal{C}\land\bigvee_{j\in J_1\cup J_2} l_j & \\
& ?\bigvee_{j\in J_1} l_j \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\
\mathcal{C}\land\bigvee_{j\in J_1} l_j & & \mathcal{C}\land\bigvee_{j\in J_2} l_j
\end{tikzcd} \tag{2}$$

There are two ways of forgetting information,

$$\begin{tikzcd}
\mathcal{C}_1\land\mathcal{C}_2 \arrow[d] & & \mathcal{C}\land\bigvee_{j\in J_1} l_j \arrow[d] \\
\mathcal{C}_1 & \text{and} & \mathcal{C}\land\bigvee_{j\in J_1\cup J_2} l_j
\end{tikzcd} , \tag{3}$$

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 Σ. Thus the branching s-program is supposed by default to solve the *Search problem for* Σ: given an assignment of the variables, find a clause which is falsified under this assignment.

The equivalence between a Res (s) refutation of Σ 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 [26] where this game was introduced for Resolution). In short, Adversary claims that Σ 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 Σ.

Lemma 1. *If a CNF ϕ has a refutation in* $\mathsf{Res}(k+1)$ *of size N, whose corresponding branching* $(k+1)$*-program has no records of covering number $\geq d$, then ϕ has a* $\mathsf{Res}(k)$ *refutation of size $2^d\cdot N$.*

Proof. In the branching program, consider a $(k+1)$-CNF record ϕ whose covering number $\leq d$ is witnessed by variable set $V' := \{v_1,\ldots,v_d\}$. Now in place of the record ϕ we expand a tree of size 2^d questioning all the variables of V'. Each evaluation of these reduces ϕ to a k-CNF that logically implies ϕ. □

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 formulas $\mathsf{Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ 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, 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 $a \in [n]$ and let $a_1 \dots a_{\log n}$ be its binary representation. Each (unary) variable $v_{i,j}$ semantically corresponds to the conjunction $(\omega^{a_1}_{i,1} \wedge \dots \wedge \omega^{a_{\log n}}_{i,\log n})$, where

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

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

$$(\omega^{1-a_1}_{i,1} \vee \dots \vee \omega^{1-a_{\log n}}_{i,\log n}) \vee (\omega^{1-b_1}_{j,1} \vee \dots \vee \omega^{1-b_{\log n}}_{j,\log n})$$

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 $(\omega^{1-a_1}_{1,1} \vee \dots \vee \omega^{1-a_{\log n}}_{1,\log n})$ 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, 24] we consider $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ formulas 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 ϵ. 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 section we explore lower bounds for $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ in $\mathsf{Res}(s)$ for $s \geq 1$, when $G \sim \mathcal{G}^n_k(p)$.

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

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

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

Lemma 3. *(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 α-transversal sets U are β-extendible;*

2. *$\mathcal{G}$ does not have a k-clique.*

Proof. Let U be an α-transversal set and σ be a β-total restriction. The probability that a vertex w is in $N_c(U)$ is $p^{\alpha k}$. Hence $w \notin N_c(U)$ with probability $(1 - p^{\alpha k})$. After σ is applied, in each block $b \in \overline{B(U)}$ 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 σ 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 α and β. 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 α-transversal sets U and $\binom{\log n}{\beta \log n} \cdot k$ possible β-total restrictions σ.

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

Notice that the last inequality holds since $k \leq \log n$. Hence the probability that there is in G no α-transeversal set U which is β-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. □

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 seting 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. Consider the following property.

Definition 2. *(Property $\mathsf{Clique}(G, s, k)$). For any s-restriction ρ, there are no Res(s) refutations of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_{\rho}$ of size less $n^{\frac{k-1}{24^2 s}}$.*

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 1. *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 $n^{\frac{k-1}{24^2 s}}$.*

Proof. For ρ the empty assignment there are no Res(s) refutations of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$ of size smaller than $n^{\frac{k-1}{24^2 s}}$. □

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 $s \leq \frac{1}{2}\log\log n$.

Theorem 1. *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. By Lemma 3, we can fix $G \sim \mathcal{G}^n_{k,\epsilon}$ such that:

1. all α-transversal sets U are β-extendible;

2. $\mathcal{G}$ does not have a k-clique.

We will prove, by induction on $s \leq \frac{1}{2}\log\log n$, that property $\mathsf{Clique}(s,k,G)$ does hold. The result then follows by Corollary 1. Lemma 4 is the base case and Lemma 5 the inductive case. □

Lemma 4. *(Base Case)* $\mathsf{Clique}(1,k,G)$ *does hold.*

Proof. Fix $\beta = \frac{3}{4}$ and $\alpha = \frac{1}{4(2+\epsilon)} \geq \frac{1}{12}$. Let ρ 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 $2^{(w(\vdash F)-w(F))^2/V(F)}$. In our case $w(F) = 2\log n$, hence the minimal size required is $\geq 2^{\frac{(\frac{k\log n}{24}-2\log n)^2}{k\log n}} = 2^{\frac{\log n(\frac{k}{24}-2)^2}{k}} = n^{\frac{(\frac{k}{24}-2)^2}{k}}$. It is not difficult to see that $\frac{(\frac{k}{24}-2)^2}{k} \geq \frac{(k-1)}{24^2}$, the result is proved. □

Lemma 5. *(Inductive Case)*

$$\mathsf{Clique}(s-1,k,G) \textit{ implies } \mathsf{Clique}(s,k,G).$$

Proof. We prove the contrapositive. Fix $\delta = 1/24^2$. Let $\zeta(s) = (1 - \frac{1}{2^{s^2+3s}})$ and $r = \frac{\delta(k-1)\log n}{s}$. Assume there is some s-restriction ρ such that there exists a $\mathsf{Res}(s)$ refutation π of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_\rho$ with size less than n^r. Notice that $n^r \leq 2^{-\log(\zeta(s))r}$. Let us call a *bottleneck*, a record $\mathcal{R}$ in π whose covering number is $\geq \delta(k-1)\log n$. In such a record it is always possible to find $r = \frac{\delta(k-1)\log n}{s}$ s-tuples of literals $T_1 = (\ell^1_1,\ldots,\ell^s_1),\ldots,T_r = (\ell^1_r,\ldots,\ell^s_r)$ so that these s-tuples are pairwise disjoint (when considered a sets of size s) such that the $\bigwedge T_i$'s are the terms of the s-DNF forming the record. By our size assumptions on π, there are $\leq n^r$ bottlenecks. Let σ be a *s-random restriction* on the variables of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_\rho$. Let us say that σ *kills a tuple* T if it sets to 0 all literals in T (notice that a record is the negation of s-DNF) and that T *survives* σ otherwise. And that σ *kills* $\mathcal{R}$ if it kills at at least one of the tuples in $\mathcal{R}$. Let Σ_i be the event that T_i survives σ and $\Sigma_{\mathcal{R}}$ the event that R survives σ. We want to prove that with high probability σ kills all bottlenecks from π. We then study upper bounds on $\Pr[\Sigma_R]$. Since $T_1,\ldots,T_r$ are tuples in $\mathcal{R}$, then $\Pr[\Sigma_R] \leq \Pr[\Sigma_1 \wedge \ldots \wedge \Sigma_r]$. Moreover $\Pr[\Sigma_1 \wedge \ldots \wedge \Sigma_r] = \prod_{i=1}^r \Pr[\Sigma_i|\Sigma_1 \wedge \ldots \wedge \Sigma_{i-1}]$.

Claim 1. *For all $i = 1,\ldots,r$, $\Pr[\Sigma_i|\Sigma_1 \wedge \ldots \wedge \Sigma_{i-1}] \leq \Pr[\Sigma_i]$.*

Proof. We will prove that $\Pr[\Sigma_i|\neg\Sigma_1 \vee \ldots \vee \neg\Sigma_{i-1}] \geq Pr[\Sigma_i]$. This gives the claim using Lemma 6 (i). We claim that for $i \neq j \in [r]$:

$$\Pr[\Sigma_i|\neg\Sigma_j] \geq \Pr[\Sigma_i] \tag{4}$$

Hence repeated applications of Lemma 6 (ii), prove that $\Pr[\Sigma_i|\neg\Sigma_1 \vee \ldots \vee \neg\Sigma_{i-1}] \geq Pr[\Sigma_i]$.

To prove Equation 4, 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^ℓ be the set of variables in T_i mentioning block ℓ. T_i is hence partitioned into $\bigcup_{\ell \in B(T_i)} T_i^\ell$ and hence the event "T_i surviving σ", can be partitioned into the sum of the events that T_i^ℓ survives to σ, for $\ell \in B(T_i)$. Denote by Σ_i^ℓ the event "T_i^ℓ survives σ" 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:

$$\begin{aligned}
\Pr[\Sigma_i|\neg\Sigma_j] &= \Pr[\exists \ell \in B(T_i) : \Sigma_i^\ell|\neg\Sigma_j] && (5)\\
&= \sum_{\ell \in B(T_i)} \Pr[\Sigma_i^\ell|\neg\Sigma_j] && (6)\\
&= \sum_{\ell \in A} \Pr[\Sigma_i^\ell|\neg\Sigma_j] + \sum_{\ell \in B} \Pr[\Sigma_i^\ell|\neg\Sigma_j] && (7)\\
& && (8)
\end{aligned}$$

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:

$$\begin{aligned}
\sum_{\ell \in B} \Pr[\Sigma_i^\ell|\neg\Sigma_j] = \sum_{\ell \in B} \Pr[\Sigma_i^\ell] && (9)\\
&& (10)
\end{aligned}$$

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:

$$\begin{aligned}
\sum_{\ell \in A} \Pr[\Sigma_i^\ell|\neg\Sigma_j] \geq \sum_{\ell \in A} \Pr[\Sigma_i^\ell] && (11)\\
&& (12)
\end{aligned}$$

Which proves the claim since:

$$\sum_{\ell \in A} \Pr[\Sigma_i^\ell] + \sum_{\ell \in B} \Pr[\Sigma_i^\ell] = \Pr[\Sigma_i] \qquad (13)$$

□

Let $\gamma = 1/2^{s+1}$. Lemma 7 below shows that, $\Pr[\Sigma_i] \leq 1 - \frac{\gamma^s}{2^{2s}} \leq \zeta(s)$, for all $i = 1, \ldots, r$. Then by the Claim,

$$\Pr[\Sigma_R] \leq \zeta(s)^r = n^{-r}.$$

Consider now the restriction $\tau = \rho\sigma$. This is a $(s-1)$-restriction on the variables of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)$. Since there are fewer than n^r bottlenecks and $\Pr[\Sigma_R] \leq n^{-r}$, then by the union bound τ is a $(s-1)$-restriction that kills all bottlenecks of π. Then, by Lemma 1, we can morph π through the restriction τ to a $\mathsf{Res}(s-1)$ refutation of $\mathsf{Bin\text{-}Clique}^{\mathsf{n}}_{\mathsf{k}}(G)\restriction_\tau$ of size $2^{\frac{\delta(k-1)\log n}{s}} \cdot 2^{-\log(\zeta(s))\frac{\delta k \log n}{s}} = n^{\frac{\delta(k-1)}{s}(1-\log(\zeta(s)))}$. But this is smaller than $n^{\frac{\delta(k-1)}{s-1}}$ and this is contradicting $\mathsf{Clique}(s-1,k,G)$.

Notice that the previous argument can be applied while $s < \frac{\gamma}{2} \cdot \log n = \frac{\log n}{2^{s+1}}$ and since $\gamma = 1/2^{s+1}$, it holds while $\log s + s + 1 < \log\log n$, which holds at $s < \frac{1}{2}\log\log n$. □

Lemma 6. *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:

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

□

Lemma 7. *Let s be an integer, $s \geq 1$, $\gamma = \frac{1}{2^{s+1}}$, and ρ be a s-random restriction. For all s-tuples S:*

$$\Pr[S \textit{ survives } \rho] \leq 1 - \frac{\gamma^s}{2^{2s}}$$

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.

We prove that $\Pr[T \text{ survives } \rho] \leq 1 - \frac{\gamma^s}{2^{2s}}$. The result follows observing that $\Pr[T \text{ survives } \rho] \geq \Pr[S \text{ survives } \rho]$.

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. □

In the sequel we will use the fact that, while $r < \frac{\gamma}{2} \cdot \log n$,

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

since, for such r, $\frac{\log n\ - \frac{r}{\gamma} + \frac{1}{\gamma}}{\log n\ -r+1} > \frac{1}{2}$.

4 The weak Pigeonhole Principle

For $n < m$, let $\mathsf{Bin\text{-}PHP}^m_n$ be the binary encoding of the (weak) Pigeonhole Principle as showed in The Introduction in Subsection 1.1.2. First notice that an analogous of Lemma 2 holds for the pigeonhole principle too.

Lemma 8. *Suppose there are Resolution refutations of* 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 ρ be a partial assignment (a restriction) to the variables of $\mathsf{Bin\text{-}PHP}^m_n$. We call ρ a *t-bit* restriction if ρ assigns t bits of each pigeon $b \in [m]$, i.e. t variables $\omega_{b,i}$ for each pigeon b. Let $v = (i, a)$ be an assignment meaning that pigeon i is assigned to hole a and let $a_1 \ldots a_{\log n}$ be the binary representation of a. We say that a restriction ρ is *consistent* with v if for all $j \in [\log n]$, $\sigma(\omega_{i,j})$ is either a_j or not assigned. We denote by $\mathsf{Bin\text{-}PHP}^m_n\restriction_\rho$, $\mathsf{Bin\text{-}PHP}^m_n$ restricted by ρ. 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 τ 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 9. *Let ρ 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 ρ), 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 ρ 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 3 (Property $\mathsf{PHP}(s,t)$)**.** *Let $s, t \geq 1$. For any t-bit restriction ρ 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 2. *Let ρ 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 σ is a $(t+1)$-bit restriction for $\mathsf{Bin\text{-}PHP}^m_n$. However, we know from Lemma 9 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 4, 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 2. *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 $\mathsf{Bin\text{-}PHP}^m_n\restriction_\rho$. We say that a tuple is *anti-perfect* if all literals come from different pigeons.

Lemma 10. *Let s be an integer, $s \geq 1$ and σ an s-bit restriction over* $\mathsf{Bin\text{-}PHP}^m_n\restriction_\rho$ *where ρ is itself some t-bit restriction over* $\mathsf{Bin\text{-}PHP}^m_n$. *Let T be an anti-perfect s-tuple of* $\mathsf{Bin\text{-}PHP}^m_n\restriction_\rho$. *Then for all s-tuples S:*

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

and so $\Pr[S \textit{ survives } \sigma] \leq 1 - \frac{1}{(\log n - t)^s 2^s} = 1 - \frac{1}{u^s 2^s}$.

Proof. A pigeon with r distinct bits contributes a factor of

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

Noting that

$$\frac{r!}{\log n\ - t \cdot \log n\ - t - 1 \cdots \log n\ - t - r + 1} > \frac{1}{(\log n\ - t)^r},$$

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

Theorem 3. *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 ρ so that there exists a $\mathsf{Res}(s)$ refutation π of $\mathsf{Bin\text{-}PHP}^m_n\restriction_\rho$ 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 $(\ell^1_1, \ldots, \ell^s_1), \ldots, (\ell^1_r, \ldots, \ell^s_r)$ so that each s-tuple is a clause in the record and no pigeon appearing in the ith s-tuple also appears in the jth s-tuple (when $i \neq j$). This important independence condition plays a key role. Now consider a random restriction that, for each pigeon, picks uniformly at random s bit positions and sets these to 0 or 1 with equal probability. The probability that the ith of the r s-tuples survives the restriction is maximised when each variable among the s describes a different pigeon (by Lemma 10) and is therefore bound above by

$$\left(1 - \frac{2^s - 1}{2^s u^s}\right)$$

whereupon

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

which is $\leq 1/e^{\frac{(2^s-1)n}{4^{\xi(s)}s!\cdot 2^s 2^t u^{\xi(s)}}} < 1/e^{\frac{n}{4^{\xi(s)+1}s!\cdot 2^t u^{\xi(s)}}}$. 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 σ is a $s+t$-bit restriction, which moreover has covering number $< \frac{n}{4^{\xi(s)}\cdot(s-1)!2^t u^{\xi(s-1)}}$. But if the remaining $\mathsf{Res}(s)$ 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

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

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

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 4. *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 3 gives ℓ 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 5. *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 9 with $t = 0$ and finds n free choices on each branch of the tree. Following the method of Riis [30], 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 sinxe we always choose a set of masimal 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

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 Π_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 $\mathsf{Un\text{-}TC\text{-}Prin}(n)$ 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 6. Let $\mathsf{Bin\text{-}TC\text{-}Prin}(n)$ 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 $\mathsf{Bin\text{-}TC\text{-}Prin}(n)$, including total comparison, can be canonically calculated from the corresponding axioms of $\mathsf{Un\text{-}TC\text{-}Prin}(n)$ as explained in Section 6 in Defintion 4. Note that the large disjunctive clauses of $\mathsf{Un\text{-}TC\text{-}Prin}(n)$, that encode the existence of the witness, disappear completely in $\mathsf{Bin\text{-}TC\text{-}Prin}(n)$.

Lemma 11. *Suppose there is a Resolution refutation of* $\mathsf{Un\text{-}TC\text{-}Prin}(n)$ *of size* $S(n)$*. Then there is a Resolution refutation of* $\mathsf{Bin\text{-}TC\text{-}Prin}(n)$ *of size at most* $n^2 \cdot S(n)$*.*

Proof. Take a decision DAG π for $\mathsf{Un\text{-}TC\text{-}Prin}(n)$ and consider the point at which some variable $v_{i,j}$ is questioned. Each node in π will be expanded to a small tree in π', which will be a decision DAG for $\mathsf{Bin\text{-}TC\text{-}Prin}(n)$. The question "$v_{i,j}$?" in π 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 π'. 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 5.1 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 "$v_{j,i}$"). Thus, π gives rise to π' of size $n^2 \cdot S(n)$ and the result follows. □

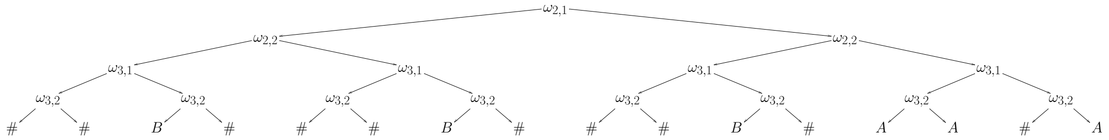

Figure 1: Example converting the question $v_{2,3}$? from a Resolution refutation of $\mathsf{Un\text{-}TC\text{-}Prin}(n)$ to a small tree in a refutation of $\mathsf{Bin\text{-}TC\text{-}Prin}(n)$. The variables $\omega_{2,1}, \omega_{2,2}, \omega_{3,1}, \omega_{3,2}$ are questioned in order. The left-hand 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).

5.2 Binary encodings of principles versus their Unary functional encodings

Recall the unary functional encoding of a combinatorial principle C, denoted $\mathsf{Un\text{-}Fun\text{-}C}(n)$, replaces the big clauses from $\mathsf{Un\text{-}C}(n)$, 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 12. *Suppose there is a Resolution refutation of* $\mathsf{Bin\text{-}C}(n)$ *of size* $S(n)$*. Then there is a Resolution refutation of* $\mathsf{Un\text{-}Fun\text{-}C}(n)$ *of size at most* $n^2 \cdot S(n)$*.*

Proof. Take a decision DAG π' for $\mathsf{Bin\text{-}C}(n)$, where w.l.o.g. n is even, and consider the point at which some variable $\nu'_{i,j}$ is questioned. Each node in π' will be expanded to a small tree in π, which will be a decision DAG for $\mathsf{Un\text{-}Fun\text{-}C}(n)$. The question "$\nu'_{i,j}$?" in π will become a sequence of questions $v_{i,1}, \ldots, v_{i,n}$ where we stop the small tree when one of these is answered true, which must eventually happen. Suppose $v_{i,k}$ is true. If the jth bit of k is 1 we ask now all $v_{i,b_1}, \ldots, v_{i,b_{\frac{n}{2}}}$, where $b_1, \ldots, b_{\frac{n}{2}}$ are precisely the numbers in $[n]$ whose jth bit is 0. All of these must be false. Likewise, if the jth bit of k is 0 we ask all $v_{i,b_1}, \ldots, v_{i,b_{\frac{n}{2}}}$, where $b_1, \ldots, b_{\frac{n}{2}}$ are precisely the numbers whose jth bit is 1. All of these must be false. We now unify the branches on these two possibilities, forgetting any intermediate information. (To give an example, suppose $j = 2$. Then the two outcomes are $\neg v_{i,1} \wedge \neg v_{i,3} \wedge \ldots \wedge \neg v_{i,n-1}$ and $\neg v_{i,2} \wedge \neg v_{i,4} \wedge \ldots \wedge \neg v_{i,n}$.) Thus, π' gives rise to π of size $n^2 \cdot S(n)$ and the result follows. □

5.3 The Ordering Principle in binary

Recall the Ordering Principle specified in Π_2 first-order logic

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

with propositional translation to the binary encoding of witnesses, $\mathsf{Bin\text{-}OP}_n$, 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 $a_1 \ldots a_{\log n}$ is the binary representation of a.

Lemma 13. $\mathsf{Bin\text{-}OP}_n$ *has refutations in Resolution of polynomial size.*

Proof. We follow the well-known proof for the unary version of the Ordering Principle, from [32]. Consider the domain to be $[n] = \{1, \ldots, n\}$. At the ith stage of the decision DAG we will find a maximal element, ordered by R, among $[i] = \{1, \ldots, i\}$. That is, we will have a record of the *special* form

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

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 $\nu_{i+1,1}, \ldots, \nu_{i+1,i}$, 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

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

Now we expand a tree questioning the sequence $w_{j,1}, \ldots, w_{j,\log n}$, 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 6. $\mathsf{Bin\text{-}OP_n}$ *has poly size resolution refutations in* $\mathsf{Res}(1)$.

$\mathsf{Bin\text{-}rLOP_n}$ is a family of contradictions based on a variant of the Ordering Principle, which is important as it exponentially separates read-once Resolution from Resolution (see [2]).

Corollary 3. $\mathsf{Bin\text{-}rLOP_n}$ *has poly size resolution refutations in* $\mathsf{Res}(\frac{1}{2}\log\log n)$.

6 Binary versus unary encodings in general

Let $\mathsf{C_n}$ be some combinatorial principle expressible as a first-order Π_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 [30] we restrict to the class of such formulas having no finite model.

Let $\mathsf{Un\text{-}C_n}$ be the standard unary (see Riis in [30]) 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 φ. 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 $\mathsf{Un\text{-}C_n}$ 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 4. *(Canonical form of* $\mathsf{Bin\text{-}C_n}$*) 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* $\mathsf{Un\text{-}C_n}$ *be its unary propositional encoding. Let* $2^{r-1} < n \leq 2^r \in \mathbb{N}$ *(*$r = \lceil\log n\rceil$*). The binary encoding* $\mathsf{Bin\text{-}C_n}$ *of* C *is defined as follows:*

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

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

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

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

Lemma 14. *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 $\mathsf{Un\text{-}C_n}$ and $\mathsf{Bin\text{-}C_n}$ be respectively the unary and binary propositional encoding. Let $n \in \mathbb{N}$. If there is a size S refutation for $\mathsf{Un\text{-}C_n}$ in $\mathsf{Res}(1)$, then there is a size S refutation for $\mathsf{Bin\text{-}C_n}$ in $\mathsf{Res}(\log n)$*

Proof. (Sketch) Where the decision DAG for $\mathsf{Un\text{-}C_n}$ 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 $\mathsf{Bin\text{-}C_n}$, and the result follows. □

As one can easily notice reading Subsection 1.1.2, 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 6. But we can easily and efficiently move between these versions in Resolution. We leave the proof to the reader.

Lemma 15. *The two versions of the binary Pigeonhole Principle (*Bin-PHP *and the one arising from Definition 6 to* PHP*) are linearly equivalent in Resolution.*

References

[1] Alekhnovich, M. Lower bounds for k-dnf resolution on random 3-cnfs. *Computational Complexity 20*, 4 (2011), 597–614. 3, 5

[2] Alekhnovich, M., Johannsen, J., Pitassi, T., and Urquhart, A. An exponential separation between regular and general resolution. *Theory of Computing 3*, 1 (2007), 81–102. 1, 4, 18

[3] Atserias, A. Improved bounds on the weak pigeonhole principle and infinitely many primes from weaker axioms. *Theor. Comput. Sci. 295* (2003), 27–39. 2

[4] Atserias, A., Bonacina, I., de Rezende, S. F., Lauria, M., Nordström, J., and Razborov, A. A. Clique is hard on average for regular resolution. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018* (2018), I. Diakonikolas, D. Kempe, and M. Henzinger, Eds., ACM, pp. 866–877. 1, 2, 8

[5] Atserias, A., Bonet, M. L., and Esteban, J. L. Lower bounds for the weak pigeonhole principle and random formulas beyond resolution. *Inf. Comput. 176*, 2 (2002), 136–152. 2, 5

[6] Beame, P., Impagliazzo, R., and Sabharwal, A. 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* (2001), IEEE Computer Society, pp. 52–68. 2

[7] Beame, P., and Pitassi, T. Simplified and improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996* (1996), IEEE Computer Society, pp. 274–282. 2

[8] Ben-sasson, E., and Wigderson, A. Short proofs are narrow - resolution made simple. In *Journal of the ACM* (1999), pp. 517–526. 2, 4, 10

[9] Beyersdorff, O., Galesi, N., and Lauria, M. A lower bound for the pigeonhole principle in tree-like resolution by asymmetric prover-delayer games. *Inf. Process. Lett. 110*, 23 (2010), 1074–1077. 1, 2, 3, 5

[10] Beyersdorff, O., Galesi, N., and Lauria, M. Parameterized complexity of dpll search procedures. *ACM Trans. Comput. Logic 14*, 3 (Aug. 2013), 20:1–20:21. 1, 2, 3, 8

[11] Beyersdorff, O., Galesi, N., Lauria, M., and Razborov, A. A. Parameterized bounded-depth frege is not optimal. *TOCT 4*, 3 (2012), 7:1–7:16. 2

[12] Bonacina, I., and Galesi, N. A framework for space complexity in algebraic proof systems. *J. ACM 62*, 3 (2015), 23:1–23:20. 2, 4

[13] Bonacina, I., Galesi, N., and Thapen, N. Total space in resolution. *SIAM J. Comput. 45*, 5 (2016), 1894–1909. 2

[14] Bonet, M. L., and Galesi, N. Optimality of size-width tradeoffs for resolution. *Computational Complexity 10*, 4 (2001), 261–276. 2, 4, 16

[15] Buss, S. R., and Pitassi, T. Resolution and the weak pigeonhole principle. In *Computer Science Logic, 11th International Workshop, CSL '97, Annual Conference of the EACSL, Aarhus, Denmark, August 23-29, 1997, Selected Papers* (1997), pp. 149–156. 1, 2, 3

[16] Dantchev, S. S., and Riis, S. Tree resolution proofs of the weak pigeon-hole principle. In *Proceedings of the 16th Annual IEEE Conference on Computational Complexity, Chicago, Illinois, USA, June 18-21, 2001* (2001), pp. 69–75. 1, 2, 3, 5

[17] Dantchev, S. S., and Riis, S. On relativisation and complexity gap. In *Computer Science Logic, 17th International Workshop, CSL 2003, 12th Annual Conference of the EACSL, and 8th Kurt Gödel Colloquium, KGC 2003, Vienna, Austria, August 25-30, 2003, Proceedings* (2003), M. Baaz and J. A. Makowsky, Eds., vol. 2803 of *Lecture Notes in Computer Science*, Springer, pp. 142–154. 5, 16

[18] Esteban, J. L., Galesi, N., and Messner, J. On the complexity of resolution with bounded conjunctions. *Theor. Comput. Sci. 321*, 2-3 (2004), 347–370. 6

[19] Filmus, Y., Lauria, M., Nordström, J., Ron-Zewi, N., and Thapen, N. Space complexity in polynomial calculus. *SIAM J. Comput. 44*, 4 (2015), 1119–1153. 2, 4

[20] Haken, A. The intractability of resolution. *Theor. Comput. Sci. 39* (1985), 297–308. 2

[21] Hrubes, P., and Pudlák, P. Random formulas, monotone circuits, and interpolation. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017* (2017), C. Umans, Ed., IEEE Computer Society, pp. 121–131. 2

[22] Krajíček, J. *Bounded arithmetic, propositional logic and complexity theory*. Cambridge University Press, 1995. 2, 6

[23] Krishnamurthy, B. Short proofs for tricky formulas. *Acta Inf. 22*, 3 (1985), 253–275. 2

[24] Lauria, M., Pudlák, P., Rödl, V., and Thapen, N. The complexity of proving that a graph is ramsey. *Combinatorica 37*, 2 (2017), 253–268. 1, 2, 3, 8

[25] Maciel, A., Pitassi, T., and Woods, A. R. A new proof of the weak pigeonhole principle. *J. Comput. Syst. Sci. 64*, 4 (2002), 843–872. 2

[26] Pudlák, P. Proofs as games. *American Mathematical Monthly* (June-July 2000), 541–550. 6, 7

[27] Raz, R. Resolution lower bounds for the weak pigeonhole principle. *J. ACM 51*, 2 (2004), 115–138. 2

[28] Razborov, A. A. Proof complexity of pigeonhole principles. In *Developments in Language Theory* (Berlin, Heidelberg, 2002), W. Kuich, G. Rozenberg, and A. Salomaa, Eds., Springer Berlin Heidelberg, pp. 100–116. 2

[29] Razborov, A. A. Resolution lower bounds for the weak functional pigeonhole principle. *Theor. Comput. Sci. 1*, 303 (2003), 233–243. 2

[30] Riis, S. A complexity gap for tree resolution. *Computational Complexity 10*, 3 (2001), 179–209. 2, 3, 4, 15, 18

[31] Segerlind, N., Buss, S. R., and Impagliazzo, R. A switching lemma for small restrictions and lower bounds for k-dnf resolution. *SIAM J. Comput. 33*, 5 (2004), 1171–1200. 2, 3, 5

[32] Stålmarck, G. Short resolution proofs for a sequence of tricky formulas. *Acta Inf. 33*, 3 (1996), 277–280. 2, 18