Resolution Lower Bounds versus the Weak Pigeonhole Principle

Noel Arteche*

Abstract

We prove that the weak pigeonhole principle $\mathrm{PHP}_n^{n^2}$ is both *necessary* and *sufficient* to derive "tight" Resolution lower bounds when working over polynomial-size constant-depth Frege systems. This addresses a problem of Pudlák, who asked whether these systems can prove Resolution lower bounds in polynomial size—a question closely related to automating propositional proof search.

We first present a new proof complexity reduction showing that $\mathrm{PHP}_n^{n^2}$ is *necessary*: any constant-depth Frege proof of a "tight" Resolution lower bound implies a proof of the weak pigeonhole principle with a polynomial overhead in size. Previous reductions relating weak pigeonhole principles and Resolution lower bounds (de Rezende et al., STOC 2021; Li et al., STOC 2026) only preserved width measures and did not scale polynomially with proof size. Our reduction also works within $\mathsf{Res}(k)$ when using the onto version of the pigeonhole principle.

Conversely, we show that $\mathrm{PHP}_n^{n^2}$ is *sufficient* over polynomial-size constant-depth Frege by formalizing tight Resolution lower bounds invoking the principle only once. Previous formalizations in this regime appealed to known quasipolynomial-size proofs of the weak pigeonhole principle but also incurred other quasipolynomial blow-ups elsewhere in the proofs.

As a consequence of our new reduction, assuming that $\mathrm{PHP}_n^{n^2}$ requires superpolynomial-size proofs in constant-depth Frege, we show that any polynomial-time algorithm automating proof search over CNF formulas in these systems yields an algorithm solving CNF-SAT in subexponential time $2^{o(n)}$ on inputs with n variables and unbounded clause width. Prior results relied on concrete cryptographic assumptions (Bonet et al., CCC 1999) or only worked for algorithms automating proof search over all constant-depth formulas rather than CNF formulas only (Papamakarios, CCC 2024).

*Lund University & University of Copenhagen, noel.arteche@cs.lth.se

Contents

1 Introduction **1**
- 1.1 Contributions . . . 3
- 1.2 Technical overview . . . 8
- 1.3 Structure of the paper . . . 13

2 Preliminaries **13**
- 2.1 Proof complexity . . . 13
- 2.2 Theories of bounded arithmetic for polynomial-size constant-depth Frege . . . 17

3 The weak pigeonhole principle is necessary **18**
- 3.1 The reduction in constant-depth Frege . . . 19
- 3.2 A framework for reductions in Res(k) . . . 26
- 3.3 The reduction in Res(k) . . . 29

4 Hardness of proving tight Resolution lower bounds **34**
- 4.1 Hardness results in constant-depth Frege . . . 35
- 4.2 Hardness in Res(k): the Iterated Lower Bounds Hypothesis . . . 36

5 Hardness of automating constant-depth Frege **39**

6 The weak pigeonhole principle is sufficient **40**
- 6.1 A first-order encoding of Ref . . . 41
- 6.2 The hard formulas and the exponential regime . . . 41
- 6.3 A subtle trick: avoiding the width lower bound . . . 42
- 6.4 The restriction argument as the weak pigeonhole principle . . . 43
- 6.5 Propositional consequences . . . 47

Acknowledgements **51**

AI disclosure **51**

References **52**

A Related results on the complexity of pigeonhole principles **57**

B Layered Resolution refutations **58**

C Stretch amplification **59**

1 Introduction

Understanding the complexity of automated theorem proving has become a central line of work in propositional proof complexity. Essential to recent progress have been deep connections between the complexity of proof search and the (in)ability of proof systems to reason about themselves. In this work, we further investigate this interplay, building on the surprising role played by Resolution size lower bounds in the study of automated proof search.

Efficient proof search in propositional systems is captured by the notion of *automatability*, introduced by Bonet, Pitassi, and Raz [BPR00]: a propositional proof system P is *automatable* if there exists an algorithm that, given a propositional tautology φ, outputs a P-proof of φ in time polynomial in the size of the shortest such proof. With the exception of tree-like Resolution, automatable in quasipolynomial time [BP96], no non-trivial proof system is known to be automatable in polynomial time. Building on a long line of work [Bus95; Iwa97; Pud03; AB04; ABMP01; AR08; MPW19], Atserias and Müller [AM20] showed that Resolution is not automatable unless **P** = **NP**—the optimal hardness assumption, as it implies the automatability of any proof system. This result was ported to a variety of other weak systems [Bel20; BY24; Gar24; GKMP20; dRGN+21; IR22; dRez21; CdR25; ADR26], those in which, informally, we know how to prove unconditional lower bounds on the lengths of proofs and the Atserias–Müller reduction could be deployed.

How hard are proof complexity lower bounds? The breakthrough of Atserias and Müller [AM20] relied on a crucial meta-mathematical insight: Resolution cannot efficiently derive any of its non-trivial proof-size lower bounds. More precisely, consider the propositional formula $\textsc{Ref}_s(\varphi)$ stating that the CNF formula φ admits a Resolution refutation of length at most s. Atserias and Müller showed that whenever φ is unsatisfiable and hence $\textsc{Ref}_s(\varphi)$ may encode a non-trivial Resolution size lower bound, Resolution itself requires length $2^{\Omega(s/n)}$ to refute this lower-bound statement. This is in contrast with the situation for satisfiable φ, when $\textsc{Ref}_s(\varphi)$ encodes a "trivial" lower bound. In this setting, Pudlák [Pud03] showed that $\textsc{Ref}_s(\varphi)$ admits refutations of polynomial size. This gap is precisely the **NP**-hardness reduction in [AM20]: if φ is satisfiable, then $\textsc{Ref}_s(\varphi)$ admits a short refutation that a hypothetical automating algorithm would find quickly; if φ is unsatisfiable, then no automating algorithm can find a refutation of $\textsc{Ref}_s(\varphi)$ within a polynomial number of steps, since no refutation of that length exists.

At the core of this argument is the following fundamental question: *When can a proof system prove its own lower bounds?* Pudlák rapidly identified this question as "even more fundamental than automatability" [Pud20, p. 3] and highlighted *constant-depth Frege* as the frontier system for it [Pud20, Problem 2]. Namely, can constant-depth Frege prove any of its own size lower bounds efficiently? If not, automating constant-depth Frege would be **NP**-hard.

Proof search in constant-depth Frege: the frontier Constant-depth Frege, often referred to as $\mathbf{AC}^0$-Frege, lies at the frontier between weak and strong proof systems. These are fragments of standard Hilbert-style propositional calculi in which all the formulas appearing in the proofs have their depth bounded by some constant. The systems are weak enough that unconditional proof-length lower bounds are known [Kra94; Ajt94; PBI93; KPW95; PRST16; Hås21; HR25; Hås25], yet strong enough for some non-trivial combinatorial reasoning, as witnessed, for example, by their quasipolynomial-size upper bounds for the weak pigeonhole principle [PWW88; Kra01; MPW02]. On both fronts, our understanding is far from satisfactory: techniques for lower bounds, almost exclusively inspired by random restriction arguments in circuit complexity, are scarce, while upper bounds are often hard to push to polynomial size.

Our understanding of proof search in constant-depth Frege systems is similarly incomplete. In the early 2000s, building on cryptographic hardness results for strong proof systems [KP98; BPR00; ACG24], Bonet, Domingo, Gavaldà, Maciel, and Pitassi [BDG+04] showed that if factoring n-bit Blum integers requires time 2^{n^δ} for some $\delta > 0$, then $\mathbf{AC}^0$-Frege systems of sufficiently large constant depth are not automatable in polynomial time. More recently, Papamakarios [Pap24], inspired by the ideas of Atserias and Müller, obtained something remarkably close to optimal: depth-d Frege is not automatable unless $\mathbf{P} = \mathbf{NP}$. We remark, however, that Papamakarios does not obtain lower bounds for the Ref formulas, but rather for some related formulas that do not encode Resolution lower bounds. These results leave room for improvement: the cryptographic assumption of [BDG+04] is quite strong, while [Pap24] only rules out automating algorithms that search for proofs of all depth-d formulas, rather than CNF formulas only.

Pudlák's problem: Can constant-depth Frege prove Resolution lower bounds? In light of this discussion, a more satisfactory hardness result for proof search in constant-depth Frege would likely first prove that these systems cannot efficiently derive their own size lower bounds. At present, this appears challenging and closely related to the longstanding problem of separating the constant-depth Frege hierarchy. As a first step, one should understand whether Resolution (i.e., depth-0 Frege) lower bounds are already hard for higher levels of the hierarchy. In particular, Pudlák asked the following question.

Problem 1.1 ([Pud20, Problem 3])**.** *Is there some depth $d \geq 1$ and a sequence of unsatisfiable CNF formulas $\{\varphi_n\}_{n\in\mathbb{N}}$ of size at most $\mathrm{poly}(n)$ such that a superpolynomial lower bound on the length of Resolution refutations of $\{\varphi_n\}_{n\in\mathbb{N}}$ can be proved in depth-d Frege using polynomial-size proofs? More precisely, is there a function $s : \mathbb{N} \to \mathbb{N}$ with superpolynomial growth such that the tautologies $\{\neg \mathrm{Ref}_{s(n)}(\varphi_n)\}_{n\in\mathbb{N}}$ have polynomial-size proofs in depth-d Frege?*

The problem remains open, with some recent progress towards a positive answer. Indeed, on the positive side, Li, Li, and Ren [LLR26] and Arteche, de Rezende, Khaniki, and Risse [AdRKR26] showed that several Resolution lower bounds are formalizable within constant-depth Frege in *quasipolynomial size*. These include, for example, Haken's exponential lower bound for the pigeonhole principle within Res(log) [LLR26] and the Atserias–Müller argument itself within depth-4 Frege [AdRKR26]. In contrast, Arteche, Atserias, de Rezende, and Khaniki [AAdRK25] constructed true Resolution lower-bound statements that are arbitrarily hard for constant-depth Frege to derive. Yet, whether polynomial-size proofs are possible for some concrete lower bound remains unclear.

For the existing formalizations [LLR26; AdRKR26], the main bottleneck preventing polynomial-size proofs—though not the only one—is the *weak pigeonhole principle* (WPHP), stating that there is no injective map from $[m]$ to $[n]$ for $m \gg n$ (say, $m = 2n$ or $m = n^2$). All of the existing formalizations in $\mathbf{AC}^0$-Frege, as well as recent attempts at formalizing even stronger lower bounds [DR26], rely on it. The principle has quasipolynomial-size proofs in $\mathbf{AC}^0$-Frege [PWW88; Kra01; MPW02], and it is used to simulate the restriction arguments needed to lift a Resolution width lower bound into a size lower bound. Improving the existing quasipolynomial-size upper bounds for WPHP would be a way towards answering Pudlák's problem in the affirmative, but this appears challenging.

Reverse mathematics of proof complexity A natural question is then whether the weak pigeonhole principle is simply *necessary* to derive any Resolution lower bound. The work of de Rezende, Göös, Nordström, Pitassi, Robere, and Sokolov [dRGN+21] gave evidence for this necessity: they showed that any low-width Resolution refutation of $\mathrm{Ref}(\varphi)$ for unsatisfiable φ implies a low-width refutation of the (retraction) weak pigeonhole principle (rWPHP). Li, Li, and Ren [LLR26] generalized this

reduction to study the *reverse mathematics of Resolution lower bounds*: what is the necessary and sufficient reasoning power to derive such lower bounds? This fits into the recent trend of *feasible reverse mathematics*, where, for example, Chen, Li, and Oliveira [CLO24] studied the necessary and sufficient axioms to derive many well-known results of circuit complexity, and similar work is being conducted in the context of algebraic complexity [GGRT26; GGL+26]. In this spirit, Li, Li, and Ren [LLR26] did the same for Resolution lower bounds and unveiled rich connections between these statements and subclasses of **TFNP**.

Unfortunately, the existing work on reverse mathematics turns out to be too coarse to answer Pudlák's question. The reductions in [dRGN+21; LLR26] are well-suited to preserving width/degree measures, but scale poorly when it comes to proof size. In particular, if $\textsc{Ref}(\varphi)$ admits a polynomial-size refutation in constant-depth Frege, then these reductions do not imply polynomial-size proofs of the related rWPHP principle.

1.1 Contributions

In this work, we study the reverse mathematics of Resolution lower bounds at a fine-grained level, where "fine-grained" means that our baseline system of reasoning is polynomial-size constant-depth Frege. In this setting, we prove that the weak pigeonhole principle with n^2 pigeons and n holes ($\mathrm{PHP}_n^{n^2}$) is both *necessary* and *sufficient* to derive a broad class of tight Resolution lower bounds.

1.1.1 The weak pigeonhole principle is necessary

The central contribution of this paper is a new proof complexity reduction that works over constant-depth Frege (and, with some tweaks, also over $\mathsf{Res}(k)$ for constant k) and shows that any proof of a Resolution lower bound implies a proof of a related instance of the pigeonhole principle. Unlike reductions mentioned above [dRGN+21; LLR26], our result is designed to keep the overhead in size bounded by a polynomial. The reduction applies to the standard unary encoding of the injective pigeonhole principle PHP_n^m, which falsely claims that there exists an injective map from $[m]$ to $[n]$ for $m > n$.

Theorem 1.2 (The new reduction, informal – see Theorem 3.2)**.** *Suppose a CNF formula φ admits a Resolution refutation of length L and depth D. For every s and every d, if there is a depth-d Frege refutation of $\textsc{Ref}_s(\varphi)$ in size T, then there is a depth-$(d+2)$ Frege refutation of $\mathrm{PHP}^L_{s/D}$ in size $\mathrm{poly}(T, L, D, s, |\varphi|)$.*

Let us plug in some parameters to make the statement more understandable. Say we take an arbitrary unsatisfiable formula φ over n variables, which we know admits a trivial refutation of length $L = 2^{n+1}$ and depth n. Then the statement above implies the following result.

Corollary 1.3. *For every unsatisfiable CNF formula φ over n variables, every $d \in \mathbb{N}$ and every $0 < \varepsilon < 1$, if depth-d Frege refutes $\textsc{Ref}_{2^{\varepsilon n}}(\varphi)$ in polynomial size, then depth-$(d+2)$ Frege refutes $\mathrm{PHP}^{2^{n+1}}_{2^{\varepsilon n}/n}$ in polynomial size.*

Observe that $\mathrm{PHP}^{2^{n+1}}_{2^{\varepsilon n}/n}$ is precisely an instance of the weak pigeonhole principle $\mathrm{PHP}^{m^{O(1)}}_m$ with polynomial stretch: it maps $\mathrm{poly}(m) = 2^{n+1}$ pigeons to $m = 2^{\varepsilon n}/n$ holes. By a simple well-known amplification argument (Proposition 4.1), this implies a refutation of the quadratic-stretch $\mathrm{PHP}^{m^2}_m$ at depth $O(d)$.

It is not known whether this weak pigeonhole principle can be refuted in polynomial size in constant-depth Frege. Let us assume for now that it cannot: suppose it requires superpolynomial-size refutations in any constant depth. Then, constant-depth Frege does not have polynomial-size proofs of any Resolution lower bound of the form $2^{\Omega(n)}$.

1.1.2 The WPHP Hypothesis

How plausible is it that the weak pigeonhole principle $\mathrm{PHP}_n^{n^2}$ is hard for $\mathbf{AC}^0$-Frege? One should be careful about making such a conjecture lightly, since nontrivial upper bounds are known. For depth-0.5 Frege, Maciel, Pitassi, and Woods [MPW02] showed an $n^{O(\log n)}$ upper bound for these formulas. Furthermore, a substantial size–depth trade-off already follows from Paris, Wilkie, and Woods [PWW88]. Writing $\log^{(d)} n$ for the d-fold iterated logarithm, for every fixed $d \geq 1$ there is a constant depth $O(d)$ at which $\mathrm{PHP}_n^{n^2}$ has proofs of size $n^{O_d(\log^{(d)} n)}$. This is asymptotically better than $n^{O(\log n)}$ as d goes to infinity. It is open whether any of these bounds is tight, or whether polynomial-size proofs exist at some higher constant depth. In light of this, we put forward the following working hypothesis.

Hypothesis 1.4 (The WPHP Hypothesis)**.** *For every $d \in \mathbb{N}$ and every large enough n, the unsatisfiable CNF formulas $\mathrm{PHP}_n^{n^2}$ require size at least $n^{\omega(1)}$ to be refuted in depth-d Frege.*

We choose to call Hypothesis 1.4 a *hypothesis* rather than a *conjecture*, as the literature shows a lack of consensus on the expected hardness of these formulas. As a matter of fact, the literature already contains varied opinions on the hardness of PHP_n^{2n}; we review them here, as they are relevant to the formulas $\mathrm{PHP}_n^{n^2}$ we are interested in. Maciel, Pitassi and Woods conjecture that "it is *not* possible to prove the weak pigeonhole principle PHP_n^{2n} with polynomial-size, small-depth (say depth 2 or 3) Frege proofs" [MPW02, §7]. However, their conjecture does not preclude polynomial-size proofs at higher depths. In his doctoral thesis, Atserias [Ats02] makes such a complementary conjecture explicit: he conjectures there exists some constant depth d such that PHP_n^{2n} admits polynomial-size proofs in depth-d Frege [Ats02, Conjecture 2]. Both of these conjectures remain open[1]. Since Atserias conjectures polynomial-size proofs for PHP_n^{2n}, if true, his conjecture would also imply short proofs of $\mathrm{PHP}_n^{n^2}$—contradicting our WPHP Hypothesis.

We see the WPHP Hypothesis as an extremely productive line of further inquiry, regardless of its ground truth, and one that is ambitious but conceivably within the reach of current techniques. In some sense, either outcome is plausible and, thanks to the results in our work, interesting for different reasons: if true, proof search in constant-depth Frege is as hard as CNF-SAT, as we show below; if false, then our own work in this paper (see Theorem 1.7) implies that Pudlák's problem has a positive answer. That would make the conjecture of Atserias more plausible; if that one holds, it would in turn imply that with sufficiently uniform proofs feasible within $\mathrm{I}\Delta_0$, this theory would prove the infinitude of primes [Wil78; PWW88], among other things [MPW02, §7].

For the reader interested in the general complexity of PHP_n^m, Appendix A surveys the known upper and lower bounds for the different pigeonhole principles in constant-depth Frege.

1.1.3 Obstacles towards Pudlák's problem

In general, the reduction in Theorem 1.2 in the parameter regime of the WPHP Hypothesis (Hypothesis 1.4) applies to a broader setting of what we call *naturally tight Resolution lower bounds* (Definition 4.3): formulas for which there exists a length-s Resolution lower bound and a length-$s^{O(1)}$ upper bound of depth $s^{1-\eta}$. For any such formula, a proof of the length-s lower bound within depth-d Frege produces, using Theorem 1.2, a proof of the weak pigeonhole principle $\mathrm{PHP}_n^{n^2}$ at depth $O(d)$.

[1] There exist recent claims of an $n^{\Omega(\log n)}$ lower bound for PHP_n^{2n} in Res(log) obtained using large language models; this was carried out by some participants of the program *Logical Foundations of Complexity Theory* at the Isaac Newton Institute of the University of Cambridge during the autumn of 2026 [JT26]. This claim has not been fully verified yet, and, in any case, the purported proof does not seem to work for quadratic stretch nor higher depth.

Corollary 1.5 (Hardness of naturally tight lower bounds – see Theorem 4.5)**.** *Assuming the WPHP Hypothesis (Hypothesis 1.4), constant-depth Frege systems cannot derive any naturally tight Resolution lower bounds in polynomial size. That is, if $\{\varphi_n\}_{n\in\mathbb{N}}$ is a sequence of unsatisfiable formulas that admits a naturally tight Resolution lower bound of length $s = s(n)$, then for every $d \in \mathbb{N}$, depth-d Frege requires superpolynomial size to prove the tautologies $\{\neg\textsc{Ref}_{s(n)}(\varphi)\}_{n\in\mathbb{N}}$ encoding such a lower bound.*

Naturally tight lower bounds include, for instance, Haken's $2^{\Omega(n)}$ lower bound for the pigeonhole principle [Hak85] or the classical lower bounds for Tseitin contradictions [Urq87], as well as many other well-known formulas that are naturally tight in our sense. As a result, if Pudlák's question (Problem 1.1) can be answered affirmatively, it must be by a loose lower bound. We leave it open whether an $n^{\Omega(\log n)}$ Resolution lower bound on PHP^{n+1}_n can be proven without WPHP—namely, without counting.

1.1.4 From automating constant-depth Frege to solving CNF-SAT

As a consequence of Corollary 1.3 and the WPHP Hypothesis, we show how to set up a version of the Atserias–Müller reduction to CNF-SAT that can be carried out in subexponential time in the context of constant-depth Frege. Suppose there exists a polynomial-time automating algorithm A for depth-d Frege. For an arbitrary CNF formula φ over n variables and $\varepsilon > 0$, construct $\textsc{Ref}_{2^{\varepsilon n}}(\varphi)$ and run A on it for a polynomial number of steps. If φ is satisfiable, then by Pudlák's construction [Pud03], there is a polynomial-size refutation of $\textsc{Ref}_{2^{\varepsilon n}}(\varphi)$ that A must find; if φ is unsatisfiable, then A cannot find a refutation in polynomial time, or else, by Corollary 1.3, this would imply polynomial-size refutations of $\mathrm{PHP}^{m^2}_m$, contradicting the WPHP Hypothesis.

In this way, for every $\varepsilon > 0$, we obtain an algorithm solving CNF-SAT in time $2^{\varepsilon n}$; our reduction is robust in that it can handle an arbitrary number of clauses of unbounded width. But we can do better: we can make each of these $2^{\varepsilon n}$-time algorithms more robust by making them output satisfying assignments when φ is satisfiable and then dovetail them in the style of Levin's universal search to construct *a single algorithm* solving CNF-SAT in time $2^{o(n)}$.

Theorem 1.6. (*From automatability to* CNF-SAT *– see Theorem 5.1*) *Assume the WPHP Hypothesis (Hypothesis 1.4). For every $d \in \mathbb{N}$, if depth-d Frege is automatable in polynomial time on CNF formulas, then* CNF-SAT *can be solved by a deterministic algorithm running in time $2^{o(n)} \cdot (n+m)^{O(1)}$, where n is the number of variables and m the number of clauses.*

The reduction above remains nontrivial even when compared against recent AI-assisted breakthroughs by OpenAI [Ope26] and Anthropic [Cla26], as no algorithm is known to solve CNF-SAT in subexponential time when the input formulas have unbounded clause width. Indeed, the new claimed algorithms show that k-SAT can be solved in time $2^{o(n)}$ for every fixed k, where k is a bound on the clause width of the CNF formulas given as input, refuting the Exponential Time Hypothesis (ETH) of Impagliazzo, Paturi and Zane [IP01; IPZ01] as well as the Strong Exponential Time Hypothesis (SETH). Follow-up expository work [ODo26; Sub26] has shown that $\mathbf{NTIME}(T) \subseteq \mathbf{TIME}(2^{o(T)})$, but this is still insufficient for subexponential-time algorithms for CNF-SAT: the input is a CNF formula of n variables and m clauses, so these results only yield algorithms where the exponent grows with m and not only with n. It is generally not known how to turn a subexponential-time algorithm solving k-SAT into one solving CNF-SAT while retaining the savings, and the hardness of CNF-SAT remains a plausible assumption underlying much of fine-grained complexity [PW10; CDL+16; dHKS15; GdHKS16]. The standard reduction from CNF-SAT to k-SAT typically produces polynomially many new variables, and Santhanam and Srinivasan [SS12] showed unconditionally that the other natural approach, sparsification, does not work for unbounded-width CNF formulas.

The best known classical algorithms for unrestricted CNF-SAT achieve only subconstant savings in the exponent, with runtime $2^{n-o(n)}$ for polynomial-size formulas [Sch05; DW05; DHW06]. In contrast, our automatability reduction yields a $2^{o(n)}$-time algorithm for unrestricted CNF-SAT.

Ignoring the WPHP Hypothesis for a moment, we improve on Bonet et al. [BDG+04] by avoiding their concrete cryptographic assumption (Diffie–Hellman); on the other hand, they can rule out the more general notion of *weak automatability* [AB04], so the results are ultimately incomparable. Similarly, ignoring the WPHP Hypothesis, we improve on Papamakarios [Pap24] by showing hardness of automatability even on CNF formulas, while he can only rule out proof search over all depth-d formulas; yet, his assumption ($\mathbf{P} \neq \mathbf{NP}$) is better than ours (CNF-SAT). Like all previous results, our theorem only applies to polynomial-time automating algorithms. To the best of our knowledge, no similar reductions are known assuming automatability in, say, quasipolynomial time.

1.1.5 The weak pigeonhole principle is sufficient

In light of Corollary 1.5, the weak pigeonhole principle is necessary for tight Resolution lower bounds. Is it also *sufficient*? Previous formalizations in this setting incurred quasipolynomial blow-ups beyond the use of the weak pigeonhole principle. For example, the proof of Haken's lower bound in [LLR26, Theorem 4.17] is carried out in $\mathsf{T}^2_2(\alpha)$, where the smash (#) function is available and allows terms to have quasipolynomial growth in the numerical values of their inputs. While smash is needed in the known proofs of the weak pigeonhole principle, the proof in [LLR26] also uses it to derive the width lower bound for PHP^{n+1}_n. Similarly, [AdRKR26, §4-5] makes heavy use of the smash function beyond WPHP and manipulates quasipolynomial-size objects at multiple points. A similar issue occurs in the work of Davis and Robere [DR26, Theorem 1.4].

In this paper, we give a simpler and more streamlined formalization of a Resolution lower bound. The entire proof goes through in polynomial-size constant-depth Frege except for one application of the weak pigeonhole principle.

Theorem 1.7 ($\mathrm{PHP}^{n^2}_n$ is sufficient – see Theorem 6.6 and 6.7)**.** *Suppose there is $d^\star \in \mathbb{N}$ such that the formulas $\{\mathrm{PHP}^{n^2}_n\}_{n\in\mathbb{N}}$ admit polynomial-size refutations in depth-$d^\star$ Frege. Then, there exists $d = d^\star + O(1)$ and a sequence of unsatisfiable formulas $\{\varphi_n\}_{n\in\mathbb{N}}$ over $\mathrm{poly}(n)$ variables and $\mathrm{poly}(n)$ clauses such that the unsatisfiable formulas $\{\mathrm{Ref}_{2^{\Omega(n)}}(\varphi_n)\}_{n\in\mathbb{N}}$ have polynomial-size refutations in depth-d Frege. Furthermore, for every $k \in \mathbb{N}$, the formulas $\{\mathrm{Ref}_{n^k}(\varphi_n)\}_{n\in\mathbb{N}}$ can also be refuted in depth-d Frege.*

In light of Theorem 1.7 directly improving the constant-depth Frege upper bound for $\mathrm{PHP}^{n^2}_n$ immediately implies a positive answer to Pudlák's problem (Problem 1.1). The formulas φ_n in the theorem are XOR-lifted instances of the graph pigeonhole principle over a constant-degree bipartite expander. Interestingly, the specific choice of formula for the purposes of this formalization is not so important. We refer the reader to the technical overview for details.

Recall that all versions of the pigeonhole principle are efficiently provable in $\mathbf{TC}^0$-Frege. As a result, Theorem 1.7 shows that Resolution lower bounds can be proved in polynomial size in those systems. Until now, polynomial-size formalizations were only known in Extended Frege [CP90; AAdRK25]. As a consequence, we also obtain that the Proof Analysis Problem (PAP) for $\mathbf{TC}^0$-Frege is **NP**-complete under polynomial-time reductions (see Corollary 6.9); this was one of the problems still open from [AAdRK25; AdRKR26].

1.1.6 Res(k) and the Iterated Lower Bounds Hypothesis

Finally, let us briefly talk about k-DNF Resolution or Res(k), a generalization of Resolution operating with k-DNF formulas instead of clauses. These systems lie in between Resolution (depth-0 Frege) and DNF Resolution (depth-1 Frege). Garlík [Gar24] adapted the Atserias–Müller technique and showed that automating Res(k) for constant k is **NP**-hard. His proof, however, avoids proving lower bounds for the Ref formulas themselves and uses a so-called k-folded variant of them. To the best of our knowledge, it is open whether Res(k) can efficiently prove any non-trivial Resolution lower bound.

The WPHP Hypothesis would rule out polynomial-size upper bounds, but the additive increase in depth in Theorem 1.2 is too coarse to say anything more specific for Res(k). For this setting, we fine-tune our reduction at the cost of using the onto version of the pigeonhole principle (ontoPHP), stating that the purported injective map is also surjective.

Theorem 1.8 (The new reduction in Res(k), informal – see Theorem 3.6)**.** *Suppose a CNF formula φ admits a Resolution refutation of length L and depth D. For every $s, k \in \mathbb{N}$, if there is a* Res(k) *refutation of* $\text{Ref}_s(\varphi)$ *in size T, then there is a* Res($2k$) *refutation of* $\text{ontoPHP}^{L}_{s/D}$ *in size* $L^{O(k)} \cdot \text{poly}(T, L, D, s, |\varphi|)$.

The hardness of the weak pigeonhole principle for Res(k) is better understood, but still far from settled. Exponential lower bounds are known for the moderately weak ontoPHP^{2n}_{n} [ABE02; SBI04; Raz15], but the hardness of $\text{PHP}^{n^2}_{n}$ remains open even for Res(2). As far as we know, the situation for WPHP in Res(k) may well be very similar to the one in Resolution. This motivates the following hypothesis.

Hypothesis 1.9 (Hardness of ontoWPHP in Res(k))**.** *For every $k \in \mathbb{N}$ there exist constants $\alpha = \alpha(k)$ and $\beta = \beta(k)$, such that $0 < \alpha < \beta$, and $n_0 \in \mathbb{N}$ such that for every $n \geq n_0$ the unsatisfiable CNF formula* $\text{ontoPHP}^{\lfloor n^{\alpha} \rfloor}_{n}$ *requires size at least $2^{n^{\beta}}$ to be refuted in* Res(k).

Our hypothesis mirrors what is known for Resolution. Indeed, Razborov's Resolution lower bound [Raz04b] of $\exp(\Omega(n^{1/3}))$ holds for the onto weak pigeonhole principle with arbitrarily many pigeons, so the hypothesis holds for $k = 1$ with constants $0 < \alpha < \beta < 1/3$. For general k, we allow both exponents to depend on k: α controls the number of pigeons, $2^{\lfloor n^{\alpha} \rfloor}$, while β controls the strength of the lower bound, $2^{n^{\beta}}$, and $\alpha < \beta$ ensures that this lower bound is superpolynomial in the size of the formula.

Under Hypothesis 1.9, the reduction in Theorem 1.8 gives an exponential lower bound for Ref formulas in Res(k) close to the $2^{\Omega(s/n)}$ bound of Atserias and Müller [AM20].

Theorem 1.10 (Resolution lower bounds hard for Res(k) – see Theorem 4.7)**.** *Assume Hypothesis 1.9. For every $k \in \mathbb{N}$ there is a constant C_k such that for every unsatisfiable formula φ over n variables, every* Res(k)*-refutation of* $\text{Ref}_s(\varphi)$ *with $s \geq n^{C_k}$ requires size* $2^{(s/n)^{\Omega(1)}}$.

Equipped with this conditional lower bound we can address the *Iterated Lower Bound Hypothesis* (ILB) of Santhanam and Tzameret [ST21]. Their goal is to generate hard sequences of formulas *for every proof system*, as follows. For a refutation system Q, consider the formula $\text{Ref}^{Q}_{s}(\varphi)$ stating that Q admits a size-s refutation of φ. Then, for a fixed formula φ and a function $s : \mathbb{N} \to \mathbb{N}$, define $\varphi_0 \coloneqq \varphi$ and $\varphi_{i+1} \coloneqq \text{Ref}^{Q}_{s(|\varphi_i|)}(\varphi_i)$. Santhanam and Tzameret put forward the hypothesis that, for any reasonable Q, if Q is not polynomially bounded, there exists some initial formula φ and a superpolynomial function s such that the sequence $\{\varphi_i\}_{i \in \mathbb{N}}$ requires size s to be refuted in Q.

Santhanam and Tzameret showed that their hypothesis holds for Resolution. We generalize this to Res(k) for all $k \in \mathbb{N}$.

Theorem 1.11. *Assume Hypothesis 1.9. For every constant $k \in \mathbb{N}$, the Iterated Lower Bounds Hypothesis holds for* $\mathsf{Res}(k)$.

1.2 Technical overview

We now present the main ideas behind our proofs. Section 1.2.1 gives an overview of the ideas behind the new reduction between Ref and WPHP, and in Section 1.2.2 we outline the new formalization making Theorem 1.7 possible.

1.2.1 The weak pigeonhole principle is necessary: the new reduction

The connection between Resolution lower bounds and weak pigeonhole principles was already present in [dRGN+21]: they showed that a Resolution refutation of $\mathrm{Ref}_s(\varphi)$ in low block-width implies a low-width refutation of the retraction weak pigeonhole principle (rWPHP). (See also [LLR26, Theorem 5.6] for a reversal in the language of bounded arithmetic.) These proofs follow a *sequential compression* argument, while our new reduction replaces this by a *parallel compression* argument.

The reduction in [dRGN+21], and why it does not work Fix φ to be an unsatisfiable CNF formula over variables $x_1, \dots, x_n$ and consider $\mathrm{Ref}_s(\varphi)$. The reduction in [dRGN+21], which we depict in Figure 1, starts from the canonical tree-like Resolution refutation τ of φ, of length $O(2^n)$ and depth n: the clauses at layer i resolve over x_i, and the leaves are the 2^n possible clauses over $x_1, \dots, x_n$, each of which weakens some clause of φ. Set $W := s/n$, and arrange the s clause-slots of $\mathrm{Ref}_s(\varphi)$—which we call *blocks*—into n layers of W blocks each. Assuming, for simplicity, an injection $f : [2W] \to [W]$ together with its inverse, one can sequentially compress τ layer by layer: the W clauses stored at one layer request $2W$ children for their premises, and f assigns these requests injectively into the W blocks of the next layer. Repeating this through all n layers yields a purported Resolution refutation using only $n \cdot W = s$ blocks. We refer to this procedure as *sequential compression*. Figure 1 depicts this process.

This reduction is well suited to preserving width measures. The variables of $\mathrm{Ref}_s(\varphi)$ are partitioned into s blocks, and the *block-width* of a clause is the number of different blocks mentioned in it. To determine the contents of one block, one follows its ancestor path through the compressed refutation until reaching the layer at which the relevant literals were fixed in τ. With a suitable binary encoding and access to f^{-1}, this takes $O(n \log W)$ queries. In this setting, [dRGN+21] shows that a Resolution refutation of $\mathrm{Ref}_s(\varphi)$ in block-width b implies a Resolution refutation of rWPHP_W^{2W} in width $O(bn \log W)$.

This very feature becomes the main obstacle when one tries to preserve *proof size*. Although one block can be determined by a decision tree of depth $q = O(n \log W)$, this tree can have size $2^{O(q)} = W^{O(n)}$; for a clause mentioning b blocks, expanding all computation paths may cost size $2^{O(qb)}$. Alternatively, since we intend to work over constant-depth Frege, we may choose to use reductions more expressive than decision trees. Yet, from the constant-depth Frege point of view, the problem is that the contents of a block are defined *sequentially*. While the child pointers themselves are local functions of f, deciding whether a literal x_j belongs to a clause at layer $k > j$ may require following $\Theta(n)$ successive pointers back to the layer where its sign was determined, so this definition is not constant-depth.

Consequently, a constant-depth Frege refutation of $\mathrm{Ref}_s(\varphi)$ of size T yields through this reduction only a pigeonhole principle refutation of size $\mathrm{poly}(|\varphi|, s, T) \cdot W^{O(n)}$. This remains problematic even

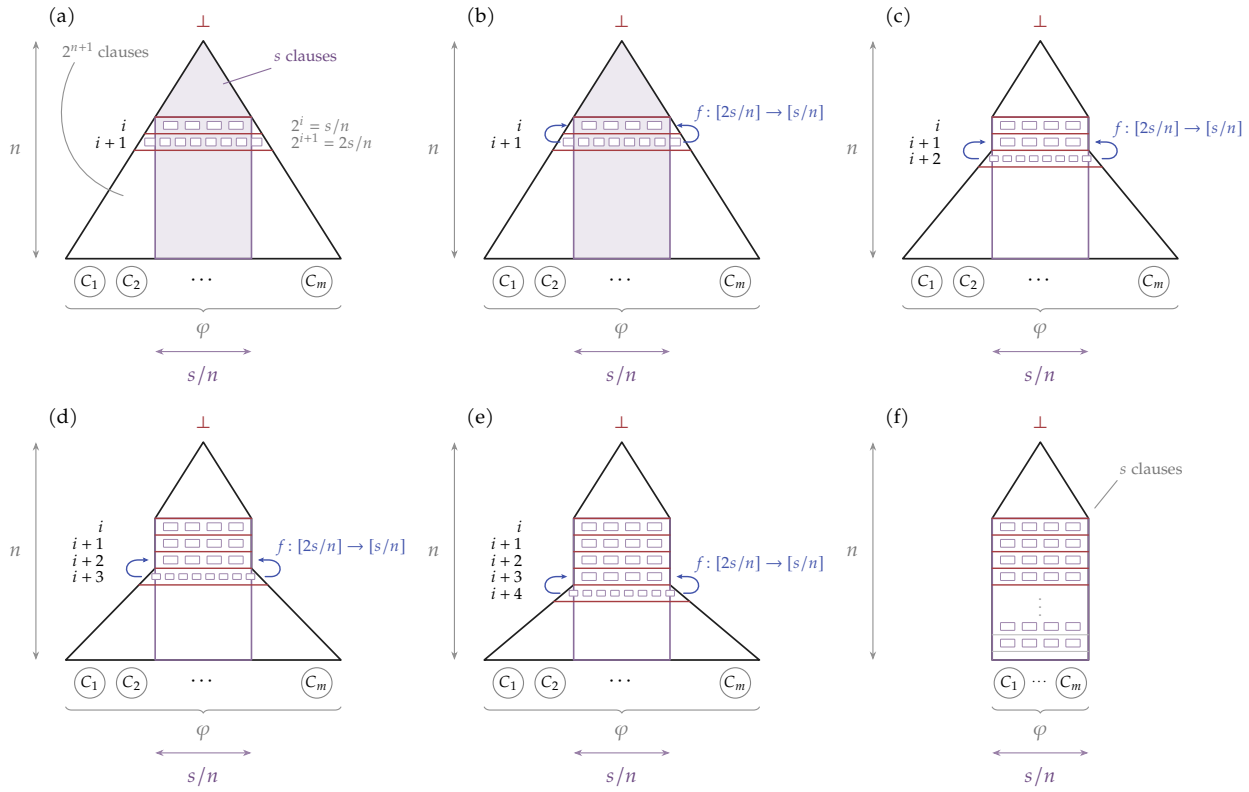


Figure 1: The *sequential compression* procedure of [dRGN+21]. The process starts with the tree-like refutation of a CNF formula $\varphi = C_1 \wedge \cdots \wedge C_m$ consisting of 2^{n+1} clauses in (a), and iteratively compresses its layers into the carved purple region consisting only of s clauses as in (f). At each iteration, the i-th layer consists of $2^i = s/n$ clauses derived from $2 \cdot 2^i = 2s/n$ clauses at layer $i+1$, and the purported injection $f : [2s/n] \to [s/n]$ is used to compress one layer at a time.

in the regime where $s = 2^{\varepsilon n}$: here $\log W = \Theta(n)$ and the overhead becomes $W^{O(\log W)} = 2^{O(n^2)}$, and depth-1.5 Frege can already refute RWPHP_W^{2W} in this size [DR26, Theorem 3.4; MPW02].

The new reduction To avoid the sequential compression argument of [dRGN+21], we introduce a different *parallel compression* mechanism. In very few words, we fix a true refutation of φ in advance and we compress it "in one shot".

Our reduction starts by fixing some Resolution refutation π of length L and depth D. The idea is now to compress the D layers of π in parallel. Consider the s available blocks of $\text{Ref}_s(\varphi)$ as arranged into D layers, each containing $W := s/D$ blocks per layer. Every layer of the refutation π, which contains up to L clauses, will be compressed into a layer consisting of W slots. To achieve this, an injective map $f : [L] \to [W]$ suffices, as seen in Figure 2.

The reduction determines the contents of blocks locally. Denote by π_k the clauses appearing on layer $k \in [D]$ in π. For a fixed clause $C \in \pi_k \subseteq \pi$, we know everything about C: its contents, the clauses at layer $k+1$ used to derive it, etc. Now, to illustrate how the rest of the reduction works, suppose we have a block $B \in [s]$ at layer k, and we want our reduction to determine whether its left premise in the Resolution step was some block $B' \in [s]$ at layer $k+1$. This information corresponds

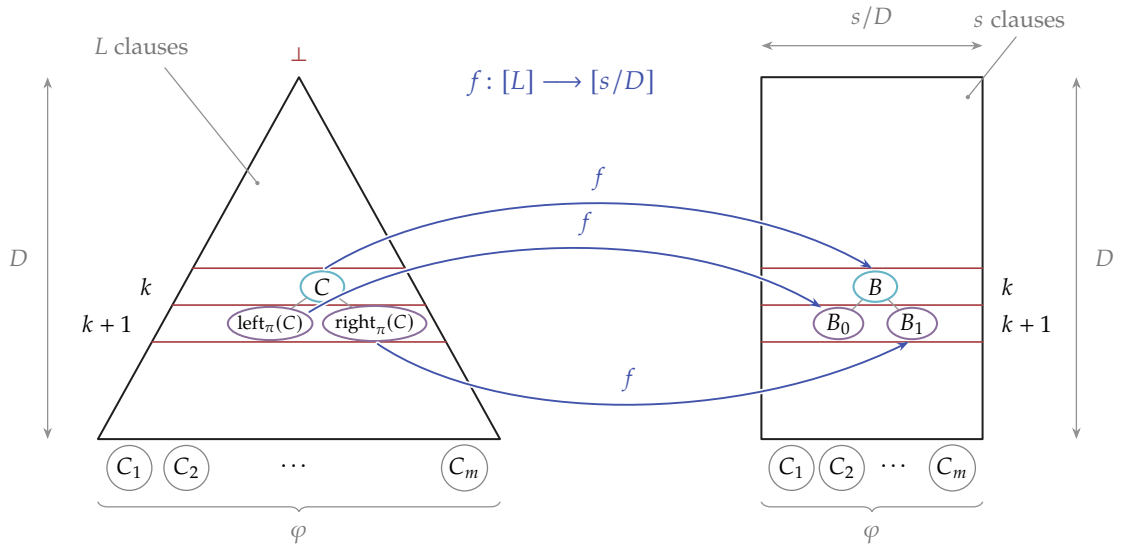


Figure 2: The new *parallel compression* reduction. On the left, a fixed Resolution refutation π for the CNF formula $\varphi = C_1 \wedge \cdots \wedge C_m$ consisting of L clauses across D layers; on the right, the compressed refutation consisting only of $s \ll L$ clauses. For every original clause $C \in \pi$, its premises $\mathrm{left}_\pi(C)$ and $\mathrm{right}_\pi(C)$ are fixed and their literal contents are known. The purported injective map $f : [L] \to [s/D]$ suffices to handle all layers "in parallel".

to the variable $\mathsf{lpoint}^B_{B'}$ of $\textsc{Ref}_s(\varphi)$. We can make the following substitution,

$$\mathsf{lpoint}^B_{B'} \leftrightarrow \bigvee_{C \in \pi_k} \left(f_{C,B} \wedge f_{\mathrm{left}_\pi(C),B'} \right). \tag{1.1}$$

where we write $\mathrm{left}_\pi(C)$ for the clause used to derive C on the left. Note that this information is fixed in advance by π. Here, $f_{C,B}$ are the variables of PHP^L_W, stating that clause C from layer k is mapped to the block B of layer k.

It turns out the definition of pointer variables above is the most complicated case in this reduction. The contents of a block are now straightforward to determine: a literal ℓ is present in block B at layer $k \in [D]$ if some clause $C \in \pi_k$ contains ℓ and f maps C to B. The variable encoding this is lit^B_ℓ, so it suffices to substitute

$$\mathsf{lit}^B_\ell \leftrightarrow \bigvee_{\substack{C \in \pi_k \\ \ell \in C}} f_{C,B}\,. \tag{1.2}$$

Once the definitions are in place, it is tedious but straightforward to verify that all the clauses of $\textsc{Ref}_s(\varphi)$ can be derived from PHP^L_W within constant-depth Frege. Theorem 1.2 then follows from implementing this idea in detail.

Some remarks on the stretch In order to apply Theorem 1.2, we need to have access to a true refutation π of length L and depth D for φ. This completely fixes the number of pigeons L in $\mathrm{PHP}^L_{s/D}$, and we can then set the desired stretch by modifying s, which determines the strength of the Resolution lower bound $\textsc{Ref}_s(\varphi)$ we are reducing to.

The result in Corollary 1.5 is obtained by choosing L to be the length of the shortest existing refutation and s to be the tight lower bound known to hold, which is polynomially related to L.

Hence, when we deal with a specific φ and we know of some specific refutation π of length L, we can get some control over the stretch of $\mathrm{PHP}^L_{s/D}$. In general, we do not have such a tight control: for the reduction to CNF-SAT in Theorem 1.6 we work with arbitrary unsatisfiable φ, without any assumption on the length of their shortest refutations. We can still compress the trivial exponential-size refutation into a smaller number $s = 2^{\varepsilon n}$ of clauses for every $\varepsilon \in (0,1)$, but the instance PHP^L_W always has polynomial stretch.

Similarities with Razborov's reduction to truth-table tautologies In hindsight, our reduction resembles the one devised by Razborov [Raz98; Raz04b] connecting circuit lower bound statements and the weak pigeonhole principle. Suppose that we are able to prove some circuit lower bound for a Boolean function $f : \{0,1\}^n \to \{0,1\}$ encoded by its truth table. In propositional logic, this is done by the so-called *truth-table tautologies*. These are formulas denoted $\mathrm{TT}(f, s)$, where the variables encode a circuit of size s computing f.

Just like every unsatisfiable CNF formula admits a trivial exponential-size refutation, for an arbitrary Boolean function f, we always know f is computable by a perfect DNF P of exponential size $O(2^n)$. Assuming an injective map $c : [2^n] \to [s]$, we can carry out a similar parallel compression argument as above: map each one of the terms of P to one of the s available gates over the variables of $\mathrm{TT}(f, s)$. It follows that if one can refute $\mathrm{TT}(f, s)$, one must also refute $\mathrm{PHP}^{2^n}_s$. Razborov showed this reduction goes through in Polynomial Calculus [Raz98] and can be adapted to Resolution in the presence of onto and functionality axioms (ontoFPHP). For the application of showing the hardness of truth-table tautologies $\mathrm{TT}(\mathrm{SAT}, n^k)$ encoding $\mathbf{NP} \not\subseteq \mathbf{P}/\mathrm{poly}$, one needs hardness of $\mathrm{ontoFPHP}^{2^n}_{n^k}$. These formulas are hard for Resolution [Raz04a; Raz04b], but not for constant-depth Frege (see Table 1).

For the reduction to truth-table tautologies to work in Resolution it is key that the substitution encoding the reduction does not increase the depth of the formulas. In our case, due to the depth-2 formulas in (1.1), the substitution immediately increases the depth. As a result, we need to assume hardness of WPHP over depth-d Frege for larger d. The exponential-stretch instances of Razborov, if hard, would improve the reduction in Theorem 1.6 to a polynomial-time reduction to SAT, since we would be able to choose s to be a fixed polynomial in n. Unfortunately, such a large stretch is easy for $\mathbf{AC}^0$-Frege.

Optimizing the reduction for $\mathsf{Res}(k)$ Although we cannot use Razborov's exponential-stretch PHP instances, we can still get some inspiration from other elements of his proofs. The naive implementation of the reduction for $\mathrm{TT}(f, s)$ does not actually stay within Resolution, since variables are assigned to clauses and not just literals. Razborov fixed this by mapping positive and negative literals of the same variable to "complementary formulas", and adding onto (and functionality) axioms to the pigeonhole principle. This is precisely what we do to fine-tune our reduction for $\mathsf{Res}(k)$ in Theorem 1.8.

Consider the substitution in (1.2). In order to avoid the increase in depth that would occur from substituting for $\neg\mathsf{lit}^B_\ell$ the exact negation $\neg \bigvee_{\substack{C \in \pi_k \\ \ell \in C}} f_{C,B} = \bigwedge_{\substack{C \in \pi_k \\ \ell \in C}} \neg f_{C,B}$, we map lit^B_ℓ to the same formula as before, but map its negation $\neg\mathsf{lit}^B_\ell$ to a different formula,

$$\neg\mathsf{lit}^B_\ell \leftrightarrow \bigvee_{\substack{C \in \pi_k \\ \ell \notin C}} f_{C,B}\,. \tag{1.3}$$

The onto axioms are added to reason precisely about the fact that these two substitutions are

complementary. Observe how $\mathsf{lit}_\ell^B \vee \neg\mathsf{lit}_\ell^B$ is now mapped exactly to $\bigvee_{C\in\pi_k} f_{C,B}$, which is an onto axiom of $\textsc{ontophp}_W^L$ for hole $B \in [W]$. In the presence of these axioms, we can easily argue about the correctness of the reduction while keeping the width of the terms bounded: a $\mathsf{Res}(k)$ refutation of $\textsc{Ref}_s(\varphi)$ now implies a refutation of the onto weak pigeonhole principle in $\mathsf{Res}(2k)$.

To handle this situation smoothly, we introduce a *general framework for substitutions within* $\mathsf{Res}(k)$ that need not map complementary literals to syntactically complementary formulas. The underlying idea goes back to the monotone translation of Buss and Pitassi [BP98] for the pigeonhole principle, and appears in more general forms in Razborov's work [Raz04a; Raz15]. We abstract this technique into a notion of *contextual substitutions*: positive and negative literals of a formula ψ are mapped to bounded-width DNF formulas that behave as complements *in the context of* some other CNF formula φ; we then show that a $\mathsf{Res}(k)$ refutation of ψ implies a $\mathsf{Res}(O(k))$ refutation of φ.

1.2.2 The weak pigeonhole principle is sufficient: the new formalization

We now turn to the converse direction: showing that the weak pigeonhole principle is essentially the only ingredient needed to formalize a Resolution lower bound. We work in the theories U^k_d-$\textsc{Ind}$ of bounded arithmetic of Beckmann, Pudlák, and Thapen [BPT14]. These can be thought of as simplified versions of the $\mathsf{T}^d_1(\alpha)$ theories of Buss [Bus86]. Roughly speaking, U^k_d-$\textsc{Ind}$ is a relativized first-order theory of arithmetic where the base language only contains functions of polynomial growth and induction is allowed for formulas consisting of a block of d bounded quantifier alternations. It will be particularly important for us that the Paris–Wilkie translation turns proofs in the theory U^1_2-$\textsc{Ind}$ into polynomial-size Resolution refutations.

The goal is to prove a Resolution lower bound explicitly isolating the use of the weak pigeonhole principle. We succeed in this by formalizing a Resolution lower bound in U^1_2-$\textsc{Ind}$ + $\mathrm{PHP}^{n^2}_n(\mathsf{U}^1_2)$; that is, Resolution reasoning expanded with substitution instances of $\mathrm{PHP}^{n^2}_n$ for U^1_2-definable relations.

The lower bounds we prove are for the formulas $\oplus\mathrm{GPHP}^{n+1}_n$: XOR-lifted instances of the graph pigeonhole principle over a constant-degree bipartite expander. The strategy is the usual one for Resolution lower bounds: we prove a linear $\Omega(n)$ width lower bound on GPHP^{n+1}_n [BW01] and then lift this to a $2^{\Omega(n)}$ size lower bound for the xorified version using a random restriction argument. (We could choose to formalize directly a size lower bound for GPHP^{n+1}_n itself, but we are not so interested in the specific formula, and this restriction argument is simpler.) It will be important for the formalization—although we can avoid this later—that we are proving an exponential lower bound, meaning that "n is in Log". This means that clauses of a purported refutation can actually be encoded as numbers inside these theories.

The fact that $n \in \mathrm{Log}$ is crucial to avoid formalizing the width lower bound itself. The reason is that the language of Beckmann, Pudlák, and Thapen for U^k_d-$\textsc{Ind}$ is deliberately extremely rich: it contains a symbol for every polynomially bounded function on natural numbers, and the base theory contains every true universal sentence over this language. We can therefore introduce a numerical function $\Lambda(C)$ measuring the minimum width needed to derive a clause C from the hard formula, with $\Lambda(C)$ bounded by $O(n)$. Instead of proving inside the weak theory that the hard formula has large Resolution width, we simply record as true universal facts that Λ is preserved under initial clauses, weakening, and Resolution steps, and that $\Lambda(\bot)$ is large; the last statement is exactly the width lower bound. The formal argument then only has to traverse the refutation looking for a wide clause, which can be done using U^1_2-induction.

The rest of the formalization corresponds to the usual restriction argument lifting width lower bounds into size lower bounds. We arrange it so that the only form of counting is one application of the weak pigeonhole principle. Similar ideas have featured before in the literature (see e.g. [Raz95;

LLR26; AdRKR26]). The argument is the usual one: we call a restriction *good* if every clause of the purported refutation is either satisfied by the restriction or becomes narrow. If no restriction were good, each restriction ρ could be associated with a pair consisting of a wide unsatisfied clause, together with a short code describing ρ relative to that clause. Distinct restrictions give distinct codes, producing an injection from a larger set of restrictions into a smaller set of such pairs. Such an injective map violates the weak pigeonhole principle. Now, once a good restriction is obtained, one proceeds through the refutation in order: the local correctness axioms of Ref and the closure properties of Λ imply that every surviving clause $C\restriction_\rho$ has $\Lambda(C\restriction_\rho) < c \cdot n$ for some constant c, whereas the final block contains $\bot$, for which $\Lambda(\bot) \geq c \cdot n$. This last step only uses U^1_2-induction, which corresponds to polynomial-size Resolution reasoning.

We remark that the relation encoding that a bad restriction ρ is mapped to some code translates into a depth-2 formula over the variables of $\mathrm{Ref}_s(\varphi)$. Hence a polynomial-size depth-d proof of the ordinary weak pigeonhole principle yields, by substitution, a polynomial-size depth-$(d+2)$ refutation of $\mathrm{Ref}_{2^{\Omega(n)}}(\varphi_n)$. In fact, with more careful analysis, the final depth is only $d + 1 + 1/2$.

While this only formalizes an exponential lower bound encoded by $\neg\mathrm{Ref}_{2^{\Omega(n)}}(\oplus\mathrm{GPHP}^{n+1}_n)$, we can easily obtain fixed-polynomial lower bounds $\neg\mathrm{Ref}_{n^k}(\oplus\mathrm{GPHP}^{n+1}_n)$ for every $k \in \mathbb{N}$ by moving into a higher depth $O(d)$. This follows from a simple restriction argument involving Ref formulas as presented in [AdRKR26, Theorem 8.3].

1.3 Structure of the paper

The paper is structured as follows. After preliminaries in Section 2, Section 3 proves the reduction from Ref to WPHP in both constant-depth Frege (Theorem 1.2) and $\mathsf{Res}(k)$ (Theorem 1.8). Section 4 shows the consequences for the hardness of Ref formulas under the WPHP Hypothesis, and Section 5 shows the reduction from automating constant-depth Frege to CNF-SAT. Finally, Section 6 proves Theorem 1.7.

2 Preliminaries

In these preliminaries we review fundamental proof complexity definitions needed in the rest of the paper. We refer to standard textbooks [Kra19; Kra95; HP93] for further details.

2.1 Proof complexity

Following Cook and Reckhow [CR79], a *propositional proof system* P for the set Taut of propositional tautologies is a polynomial-time function $P : \{0,1\}^* \to \mathrm{Taut}$ whose range is exactly Taut. If $P(\pi) = \varphi$, then we say π is an *P-proof* of φ. If φ is an unsatisfiable formula and π is an P-proof of the tautology $\neg\varphi$, then we say that π is an *P-refutation* of φ. In this paper, we mainly see the systems we work with as refutational systems.

2.1.1 Resolution

Resolution (Res) is a refutation system for CNF formulas. Accordingly, we often write $\pi : \mathsf{Res} \vdash \neg\varphi$ to express that π is a Resolution refutation of φ.

A *literal* is a propositional atom or its negation, a *clause* is a disjunction of literals, and we see clauses as sets of literals. We write $C \subseteq D$ to express that C is a subclause of D. A *Resolution refutation* of an unsatisfiable CNF formula $\varphi = C_1 \wedge \cdots \wedge C_m$ over variables $x_1, \ldots, x_n$ is a sequence $D_1, \ldots, D_s$ of clauses over $x_1, \ldots, x_n$ such that $D_s = \bot$, denoting the empty clause, and for every

$i \in [s]$, the clause D_i is either (a) one of the clauses $C_1, \ldots, C_m$ of φ, or (b) a *weakening* of a previous clause, meaning that $D_i \supseteq D_j$ for some $1 \leq j < i$, or (c) has been obtained from two previous clauses $D_j = A \vee x$ and $D_k = B \vee \neg x$, for $j, k < i$, by an application of the *Resolution rule*:

$$\frac{A \vee x \qquad B \vee \neg x}{A \vee B} \quad \text{(Res)}$$

where we say that $A \vee B$ is obtained by *resolving* over x. For the sake of completeness with later definitions, we also allow the logical axiom $x \vee \neg x$ for any variable x.

The *length* of π, denoted $\text{length}(\pi)$, is precisely s, while we denote by $|\pi|$ the length of π as a binary string, which we call its *size* and where $\text{length}(\pi) \leq |\pi| \leq (n \cdot \text{length}(\pi))^{O(1)}$. We write $\varphi \vdash^{s}_{\mathsf{Res}} \bot$ to indicate that there exists a Resolution refutation of φ in length at most s. We denote by $\text{Vars}(\varphi)$ the set $\{x_1, \ldots, x_n\}$ of variables of φ, and by $\text{Lit}(\varphi)$ the set $\{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$ of literals over those variables. Resolution is *implicationally complete*, meaning that for every pair of CNF formulas φ and ψ, if $\varphi \models \psi$, then there exists a derivation of each clause of ψ from the clauses of φ. We write $\varphi \vdash_{\mathsf{Res}} \psi$ to express that such a collection of derivations exists. The *width* of a clause $\text{w}(C)$ is the number of literals in C, and the width of a refutation π is naturally defined as $\text{w}(\pi) \coloneqq \max_{C \in \pi} \text{w}(C)$. We denote by $\text{w}_{\mathsf{Res}}(\varphi)$ the minimum width over all Resolution refutations of φ. To every Resolution refutation π we can associate a directed acyclic graph in a natural way, with the root labeled by the empty clause $\bot$. The number of vertices in this graph is exactly $\text{length}(\pi)$, and the *depth* of π, denoted $\text{depth}(\pi)$, refers to the length of the longest root-to-leaf path in π.

2.1.2 Frege systems and the cedent calculus PK

A *Frege system* [CR79] consists of a finite set of axiom schemas and inference rules that are sound and implicationally complete for the language of propositional tautologies built from the Boolean connectives negation ($\neg$), conjunction ($\wedge$), and disjunction ($\vee$). A *Frege proof* is a sequence of formulas where each formula is obtained by either substitution of an axiom schema or by application of an inference rule on previously derived formulas. The specific choice of rules does not affect proof size up to polynomial factors, as long as there are only finitely many rules and these are sound and implicationally complete.

In this paper we are concerned with the $\mathbf{AC}^0_d$-Frege systems, known as *constant-depth Frege systems*, in which lines are restricted to be Boolean formulas of unbounded fan-in and depth at most $d \in \mathbb{N}$. For the sake of precision, most of our formal statements about constant-depth Frege systems are stated for the specific Tait-style calculus PK defined below, although they could be phrased equally well in a Gentzen-style sequent calculus formalism like LK (see [Kra19, §3.4]). We follow here the presentation of Beckmann, Pudlák, and Thapen [BPT14, §1.1], but see also [Bus04; BB05; BB19].

Formulas in the refutation system PK are formed from literals together with conjunctions and disjunctions of unbounded fan-in (denoted $\bigwedge$ and $\bigvee$). In particular, negations only appear at the atomic level, and we use $\neg\varphi$ for a formula φ as an abbreviation for the corresponding formula where negations are pushed to the atoms, interchanging connectives where needed.

For such a formula φ, its *depth* is defined recursively to be $\text{depth}(\ell) \coloneqq 0$ for every literal ℓ, and $\text{depth}(\bigwedge_{i=1}^{k} \varphi_i) = \text{depth}(\bigvee_{i=1}^{k} \varphi_i) \coloneqq 1 + \max_{i \in [k]} \text{depth}(\varphi_i)$. A *cedent* is a sequence of formulas, separated by commas, interpreted as their disjunction. The system PK consists of two introduction rules and an excluded middle axiom for atoms,

$$\frac{\Gamma, \varphi \qquad \Gamma, \psi}{\Gamma, \varphi \wedge \psi} \quad (\wedge\text{-Intro}) \qquad \frac{\Gamma, \varphi, \psi}{\Gamma, \varphi \vee \psi} \quad (\vee\text{-Intro}) \qquad \frac{}{p, \neg p} \quad \text{(Atomic LEM)}$$

together with weakening and a cut rule that generalizes the Resolution rule,

$$\frac{\Gamma}{\Gamma, \Delta}\ \text{(Weak)} \qquad \frac{\Gamma, \varphi_1 \wedge \cdots \wedge \varphi_m \qquad \Gamma, \neg\varphi_1, \ldots, \neg\varphi_m}{\Gamma}\ \text{(Cut)}$$

where Γ and Δ are possibly empty cedents. Now, consider a set Φ of cedents, each interpreted as a disjunction of formulas. A *PK-derivation* of a cedent Γ from such a set Φ is a sequence of cedents ending in Γ, such that each cedent in the sequence is either in Φ, or follows from earlier cedents in the sequence by one of the rules above. A PK-*refutation* of a set of cedents Φ is a PK-derivation of the empty cedent $\bot$ from Φ. For a CNF formula φ, we consider its representation as a conjunction of disjunctions, and collect these into a set Φ; we then write $\mathsf{PK} \vdash \neg\varphi$ to indicate that PK refutes Φ. As in Resolution, the *length* of a derivation is the number of inference rules employed, while its *size* refers to the size of a binary string representing the derivation.

For every $d \in \mathbb{N}$, we write PK_d for the subsystem of PK in which every formula in a cedent has depth at most d. We also define the systems PK_d^k for $k \in \mathbb{N}$ to be the systems in which the formulas have depth $d+1$, but where all gates at depth d have fan-in at most k. The system PK_0 is readily seen to be precisely Resolution (Res). For $k \geq 2$, the system PK_0^k is denoted by Res(k) and known as *k-DNF Resolution* [Kra01]. Such a system can be seen as a generalization of Resolution that operates with *k-DNF formulas*, disjunctions of *k-terms*, which are in turn conjunctions of at most k literals. A clause is simply a 1-DNF and we refer to k as the *width* of a DNF formula. We say that a k-DNF formula is *monotone* or *positive* if it does not contain any negations.

Finally, for every $d \in \mathbb{N}$, we denote by $\mathsf{PK}_{d+1/2}$ the subsystem of PK in which a refutation π has size S if and only if every cedent has size at most S and the refutation has depth $d+1$ but every formula at depth $d+1$ has fan-in at most $\log S$. This follows the notation $\mathsf{LK}_{d+1/2}$ of Krajíček [Kra01]. The system $\mathsf{PK}_{0+1/2}$ is denoted Res(log). We sometimes use the notation $\mathbf{AC}^0_d$-Frege to refer to PK_d.

We shall also consider a variant of substitutions in which the domain is the set of literals over the variables $x_1, \ldots, x_n$ and where literals x_i and $\neg x_i$ can be respectively mapped to formulas ψ_i and χ_i where ψ_i might not be logically equivalent to $\neg\chi_i$; see Section 3.2 for details. Despite these not being technically substitutions, we often refer to them as such when there is no risk of confusion.

We also state here a useful technical lemma to make some of our Res(k) derivations easier.

Lemma 2.1 (Product rule)**.** *The* Res(k) *system can simulate the following sound inference rule using only m applications of $\wedge$-introduction and m weakenings,*

$$\frac{\Delta \vee A \qquad \Delta \vee \bigvee_{i=1}^m B_i}{\Delta \vee \bigvee_{i=1}^m (A \wedge B_i)}\ \text{Product}$$

where Δ is a k-DNF formula and the width of each term $A \wedge B_i$ is at most k.

Proof. The rule is clearly sound, so it suffices to show that Res(k) can simulate it efficiently. For $i \in [m]$, define

$$L_i := \Delta \vee (A \wedge B_1) \vee \cdots \vee (A \wedge B_i) \vee B_{i+1} \vee \cdots \vee B_m, \tag{2.1}$$

and let $L_0 := \Delta \vee B_1 \vee \cdots \vee B_m$, which is the second premise.

By induction, assume that L_{i-1} has been derived, where $1 \leq i \leq m$. Thus we have

$$\Delta \vee (A \wedge B_1) \vee \cdots \vee (A \wedge B_{i-1}) \vee B_i \vee B_{i+1} \vee \cdots \vee B_m. \tag{2.2}$$

From the first premise $\Delta \vee A$, by weakening, derive

$$\Delta \vee (A \wedge B_1) \vee \cdots \vee (A \wedge B_{i-1}) \vee A \vee B_{i+1} \vee \cdots \vee B_m. \tag{2.3}$$

Applying $\wedge$-introduction gives

$$\Delta \vee (A \wedge B_1) \vee \cdots \vee (A \wedge B_{i-1}) \vee (A \wedge B_i) \vee B_{i+1} \vee \cdots \vee B_m, \tag{2.4}$$

which is precisely L_i. This proves the validity of the rule. □

2.1.3 The Ref formulas

Fix a propositional CNF formula φ and a size parameter $s \in \mathbb{N}$. Throughout the paper we study the CNF formulas $\text{Ref}_s(\varphi)$ stating that φ has a Resolution refutation consisting of at most $s+1$ clauses. Several alternative encodings of these formulas exist in the literature (see [AAdRK25, §2.4] for a discussion). In this work, we use the natural and robust unary encoding with enabling variables used in [AM20], as presented in [AAdRK25; AdRKR26]. We remark, however, that our results also apply to the unary encoding without enabling variables (the so-called unrelativized formula of [AM20; Gar19]).

The variables of $\text{Ref}_s(\varphi)$ are partitioned into $s+1$ so-called *blocks*, each representing a clause in a purported Resolution refutation. More precisely, blocks indexed from 1 to s represent clauses in the refutation, and block $s+1$ represents the empty clause at the end of the refutation. Each block has variables representing the literals present in that purported clause, how it is obtained (either resolved or weakened from an axiom), and it contains *pointer variables* to indicate from which blocks it is derived. The formula also includes *enabling variables* used to deactivate blocks that are not used in the purported refutation, although, as mentioned, the presence of these variables is inessential to the results in this paper.

The variables of $\text{Ref}_s(\varphi)$ We assume φ is a CNF formula over n variables $x_1, \ldots, x_n$ and m clauses, and let $\text{Lit}_n := \{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$. We define the following variables with their corresponding meaning,

$\mathsf{a\text{-}lit}_\ell^A$	:	literal $\ell \in \text{Lit}_n$ is present in the clause $A \in [m]$ of φ;
enable^B	:	block $B \in [s+1]$ is enabled;
$\mathsf{derived}^B$	:	block $B \in [s+1]$ is obtained by a Resolution step;
weak_A^B	:	block $B \in [s+1]$ is obtained by weakening from clause $A \in [m]$ of φ;
lit_ℓ^B	:	literal $\ell \in \text{Lit}_n$ is present in block $B \in [s+1]$;
$\mathsf{res}_{x_i}^B$	:	block $B \in [s+1]$ is obtained by resolving over the variable x_i;
$\mathsf{lpoint}_{B'}^B$	:	block $B \in [s+1]$ is resolved on the left from block $B' \in [s+1]$, $B' < B$;
$\mathsf{rpoint}_{B'}^B$	:	block $B \in [s+1]$ is resolved on the right from block $B' \in [s+1]$, $B' < B$.

Building on these variables, the $\text{Ref}_s(\varphi)$ formula is defined as the conjunction of the following clauses.

The clauses of $\text{Ref}_s(\varphi)$

$$\left(\mathsf{enable}^B \wedge \mathsf{res}_{x_i}^B \wedge \mathsf{lpoint}_{B'}^B \wedge \mathsf{lit}_\ell^{B'}\right) \to \mathsf{lit}_\ell^B \quad \text{for } B, B' \in [s+1], B' < B, i \in [n], \ell \in \text{Lit}_n \setminus \{x_i\}, \tag{Ref-1}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{res}_{x_i}^B \wedge \mathsf{rpoint}_{B'}^B \wedge \mathsf{lit}_\ell^{B'}\right) \to \mathsf{lit}_\ell^B \quad \text{for } B, B' \in [s+1], B' < B, i \in [n], \ell \in \text{Lit}_n \setminus \{\neg x_i\}, \tag{Ref-2}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{weak}^B_A \wedge \text{a-}\mathsf{lit}^A_\ell\right) \to \mathsf{lit}^B_\ell \qquad \text{for } B \in [s+1], A \in [m], \ell \in \mathrm{Lit}_n, \tag{Ref-3}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{derived}^B\right) \to \bigvee_{i \in [n]} \mathsf{res}^B_{x_i} \qquad \text{for } B \in [s+1], \tag{Ref-4}$$

$$\left(\mathsf{enable}^B \wedge \neg\mathsf{derived}^B\right) \to \bigvee_{A \in [m]} \mathsf{weak}^B_A \qquad \text{for } B \in [s+1], \tag{Ref-5}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{derived}^B\right) \to \bigvee_{\substack{B' \in [s+1] \\ B' < B}} \mathsf{lpoint}^B_{B'} \qquad \text{for } B \in [s+1], \tag{Ref-6}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{derived}^B\right) \to \bigvee_{\substack{B' \in [s+1] \\ B' < B}} \mathsf{rpoint}^B_{B'} \qquad \text{for } B \in [s+1], \tag{Ref-7}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{lpoint}^B_{B'}\right) \to \mathsf{enable}^{B'} \qquad \text{for } B, B' \in [s+1], B' < B \tag{Ref-8}$$

$$\left(\mathsf{enable}^B \wedge \mathsf{rpoint}^B_{B'}\right) \to \mathsf{enable}^{B'} \qquad \text{for } B, B' \in [s+1], B' < B \tag{Ref-9}$$

$$\neg\mathsf{lit}^{s+1}_\ell \qquad \text{for } \ell \in \mathrm{Lit}_n, \tag{Ref-10}$$

$$\mathsf{enable}^{s+1}. \tag{Ref-11}$$

Remark 2.2 (Number of variables). The formula $\mathrm{Ref}_s(\varphi)$ has $N = \Theta(s^2 + sm + sn + mn)$ variables and $M = \Theta(s^2n^2 + smn)$ clauses. When the a-lit variables are restricted to encode a k-CNF formula over n variables and $s = n^c$ for some constant $c \geq 1$, we have $m = O(n^k)$ and $N = O(n^{\max\{2c, c+k\}})$. For most purposes, $s \geq n$ and $m = n^{O(1)}$, and thus the number of variables of $\mathrm{Ref}_s(\varphi)$ is always bounded above by $s^{O(1)}$. ◀

The following theorem, due to Pudlák [Pud03], shows that whenever φ is satisfiable, refuting $\mathrm{Ref}_s(\varphi)$ is easy even for Resolution itself. A more detailed presentation of the proof can be found in [AM20, Lemma 11], and a more approachable treatment is given in [AAdRK25, §7].

Theorem 2.3 (Pudlák's construction [Pud03, Theorem 4.1]). *There exists a polynomial-time algorithm P such that for every satisfiable CNF formula φ over n variables and m clauses satisfied by $\alpha \in \{0,1\}^n$ and every $s \geq 1$, the algorithm $P(\varphi, \alpha, 1^s)$ outputs a Resolution refutation of $\mathrm{Ref}_s(\varphi)$ of length $O(s(m + sn^2))$.*

2.1.4 Automatability

Following Bonet, Pitassi, and Raz [BPR00], for a function $T : \mathbb{N} \to \mathbb{N}$, we say that a proof system P is *automatable in time T* if there exists an algorithm that given a propositional tautology φ, outputs a P-proof of φ in time $T(|\varphi| + \mathrm{size}_P(\varphi))$, where $\mathrm{size}_P(\varphi)$ denotes the minimum size of a P-proof of φ. The definition for refutation systems and unsatisfiable φ is analogous. When T is a polynomial, we simply say P is *automatable*.

2.2 Theories of bounded arithmetic for polynomial-size constant-depth Frege

Our formalizations of Resolution lower bounds in constant-depth Frege are carried out in a family of theories of bounded arithmetic defined by Beckmann, Pudlák, and Thapen [BPT14], and reminiscent of the relativized $\mathsf{T}^i_1(\alpha)$ theories of Buss [Bus86]. The modification is intended to avoid complications coming from sharply bounded quantifiers, and has the added benefit that we have very explicit bounds on depth for a Paris–Wilkie-style propositional translation. Here we follow closely the presentation in [BPT14, §1.2].

For every finite arity $r \in \mathbb{N}$, a function $f : \mathbb{N}^r \to \mathbb{N}$ is *polynomially bounded* if there is a polynomial p such that $f(x) \leq p(x)$ for all $x \in \mathbb{N}^r$. We define $\mathcal{L}$ to be the language consisting of constant symbols 0 and 1 and, for every $r \in \mathbb{N}$, a function symbol for every polynomially bounded function $f : \mathbb{N}^r \to \mathbb{N}$ and a relation symbol for every relation on $\mathbb{N}^r$. We denote by BASE the set of all true universal $\mathcal{L}$-sentences over the standard model $\mathbb{N}$. For our formalizations, we employ an extended language $\mathcal{L}^+$ expanding $\mathcal{L}$ by a finite number of new relation symbols of finite arity. When a formula does not contain any occurrences of relation symbols in $\mathcal{L}^+ \setminus \mathcal{L}$, we say the formula is *numerical*.

We use the notation $\forall x < t(y)\, \phi$, where t is an $\mathcal{L}$-term, to denote the formula $\forall x\, (x < t(y) \to \phi)$, where the quantifier is called *bounded* (with a dual definition for existential quantifiers). Note that, due to our choice of language, an $\mathcal{L}$-term t is always some function $f \in \mathcal{L}$, and these are polynomially bounded, so bounded quantifiers in this framework always impose polynomial bounds. For $d \in \mathbb{N}$, a *strict* U_d*-formula* consists of d alternating blocks of bounded quantifiers, starting with a universal quantifier, followed by a quantifier-free $\mathcal{L}^+$-formula such that, if d is odd, the matrix is in CNF, and in DNF if d is even. A U_d*-formula* is a subformula of a strict U_d-formula. We can analogously define (*strict*) E_d*-formulas* when the first bounded quantifier is existential. If, furthermore, the matrix is a k-CNF (respectively, a k-DNF) where the width k only counts atomic literals over relation symbols in $\mathcal{L}^+ \setminus \mathcal{L}$, we talk about (strict) U^k_d-formulas (respectively, E^k_d-formulas).

For $d \in \mathbb{N}$, the theory U_d-IND is the theory BASE expanded with instances of the *induction schema*

$$\forall x\ \big(\phi(0) \wedge \forall y < x\, (\phi(y) \to \phi(y+1)) \to \phi(x)\big)\,. \tag{IND$_\phi$}$$

for every U_d-formula ϕ with parameters. The theory E_d-IND is defined analogously, and so are U^k_d-IND and E^k_d-IND. We shall also use the fact that the theories U^k_d-IND and E^k_d-IND are equivalent, meaning that their semantic consequences are the same; see [BPT14, Lemma 1.1].

The entire purpose of these theories for us is to carry our formalizations in U^k_{d+2}-IND that can be turned into PK^k_d refutations, based on a variation of the Paris–Wilkie translation [PW85]. For an $\mathcal{L}^+$ formula ϕ and an assignment α to its variables over $\mathbb{N}$, we define a PK-formula $\langle\phi\rangle_\alpha$ where $\mathcal{L}$-formulas get evaluated under α, atomic formulas on new relation symbols in $\mathcal{L}^+ \setminus \mathcal{L}$ translate into propositional atoms, and universal and existential quantifiers translate into the connectives $\bigwedge$ and $\bigvee$, respectively. We refer the reader to [BPT14, Definition 1.2] for a formal rendering of the definition.

Theorem 2.4 (Paris–Wilkie translation for U_{d+2}-IND [BPT14, Theorem A.2 and Theorem A.6])**.** *Let $d, k \in \mathbb{N}$, $k \geq 1$, and let $\phi_1(x), \ldots, \phi_\ell(x)$ be strict U^k_{d+2}-formulas with x the only free variable, such that*

$$\mathsf{U}^k_{d+2}\text{-IND} \vdash \forall x\, \neg(\phi_1(x) \wedge \cdots \wedge \phi_\ell(x))\,.$$

It holds that for every large enough $n \in \mathbb{N}$, the set of disjunctions

$$\Phi_n \coloneqq \langle\phi_1\rangle^d_{x\mapsto n} \cup \cdots \cup \langle\phi_\ell\rangle^d_{x\mapsto n}$$

admits polynomial-size refutations in PK^k_d.

3 The weak pigeonhole principle is necessary

In this section we give the reduction showing that Resolution lower bounds imply certain instances of the weak pigeonhole principle. Let us recall that by the *pigeonhole principle* we refer to the CNF

formulas of the form PHP_n^m for $m, n \in \mathbb{N}$. These are formulas over variables $p_{i,j}$ for $i \in [m]$ and $j \in [n]$ consisting of the clauses

$$\bigvee_{j \in [n]} p_{i,j} \qquad \text{for all } i \in [m], \tag{PHP-1}$$

$$\neg p_{i,j} \vee \neg p_{i',j} \qquad \text{for all } i, i' \in [m], i \neq i' \text{ and } j \in [n]. \tag{PHP-2}$$

We refer to clauses of type (PHP-1) as *totality axioms* or *pigeon axioms*, and to clauses of type (PHP-2) as *injectivity axioms* or *hole axioms*. We shall later also consider the same principle with additional *onto axioms*, referring to the clauses

$$\bigvee_{i \in [m]} p_{i,j} \quad \text{for all } j \in [n]. \tag{PHP-3}$$

In this case, the formula is denoted by $\textsc{ontophp}_n^m$.

3.1 The reduction in constant-depth Frege

For the rest of this section, let φ be an unsatisfiable CNF formula over n variables and m clauses, and fix π to be a Resolution refutation of φ consisting of L clauses and depth at most D. The reader may find it helpful to keep Figure 2 in mind when parsing the following definitions.

We assume for simplicity that every path in the refutation dag of π has length exactly D. As a consequence, the refutation π is *layered*, meaning that a clause at depth k is always resolved or weakened from clauses appearing exactly at depth $k+1$ (see Proposition B.1 for the details on how to make an arbitrary refutation layered). We denote by π_k the set of clauses appearing at depth $k \in [D]$ in the refutation, and for a clause $C \in \pi_k$, denote by $\mathrm{left}_\pi(C)$ the clause $C' \in \pi_{k+1}$ used to derive it on the left, containing the positive pivot literal, and by $\mathrm{right}_\pi(C)$ the clause used to derive it on the right, containing the negative pivot literal. Whenever a clause C is obtained by weakening some clause $C_0 \subseteq C$ where C_0 is not an axiom of φ, we assume this can be encoded by taking $\mathrm{left}_\pi(C) = \mathrm{right}_\pi(C) = C_0$. We shall also implicitly use the fact that there is an injection from each layer π_k of π into an initial segment of $[L]$; in this way, every $C \in \pi_k$ is assigned some number between 1 and L.

Let $W \in \mathbb{N}$ be a parameter that we fix later, and consider now the following depth-2 formulas over the variables $p_{C,b}$ of PHP_W^L for $C \in [L]$ and $b \in [W]$. The idea, as discussed in Section 1.2.1, is to use the purported injective map given by the PHP_W^L variables to "compress" the fixed refutation π of length L into a refutation of length $W \cdot D \ll L$. For every $k \in [D]$, $b \in [W]$, $i \in [n]$, $A \in [m]$, and $\ell \in \mathrm{Lit}_n := \{x_1, \dots, x_n, \neg x_1, \dots, \neg x_n\}$, consider the following depth-2 formulas, defined as

$$E_{k,b} := \bigvee_{C \in \pi_k} p_{C,b} \tag{3.1a}$$

$$P^{\mathrm{left}}_{k,b,c} := \bigvee_{C \in \pi_k} \left(p_{C,b} \wedge p_{\mathrm{left}_\pi(C),c}\right) \tag{3.1b}$$

$$P^{\mathrm{right}}_{k,b,c} := \bigvee_{C \in \pi_k} \left(p_{C,b} \wedge p_{\mathrm{right}_\pi(C),c}\right) \tag{3.1c}$$

$$L_{k,b,\ell} := \bigvee_{\substack{C \in \pi_k \\ \ell \in C}} p_{C,b} \tag{3.1d}$$

$$W_{k,b,A} := \bigvee_{\substack{C \in \pi_k \\ C \text{ weakens } A}} p_{C,b} \tag{3.1e}$$

$$R_{k,b,i} := \bigvee_{\substack{C \in \pi_k \\ C \text{ resolves } x_i}} p_{C,b} \tag{3.1f}$$

Now, for every $s \in \mathbb{N}$, set $W := \lfloor s/D \rfloor$, and consider the $s+1$ blocks of $\mathrm{Ref}_s(\varphi)$ arranged into

D layers, with exactly W blocks per layer, any remaining blocks from the flooring operation on W collected in the last layer, and one additional block for the empty clause of the refutation. We use the notation $B_{k,b} \in [s+1]$ for $k \in [D]$ and $b \in [W]$ to refer to the block appearing at layer k and position b in this grid arrangement. The distinguished block used to encode the empty clause $C_\emptyset = \bot$ of π is mapped to position $B_{0,0}$.

We now define a formula substitution $\sigma_{\pi,s}$ on non-root blocks $B_{k,b}$, where $k \in [D]$ and $b \in [W]$, defined as

$$\sigma_{\pi,s}(\mathsf{enable}^{B_{k,b}}) := E_{k,b} \tag{3.2a}$$

$$\sigma_{\pi,s}(\mathsf{lpoint}^{B_{k,b}}_{B_{k+1,c}}) := P^{\text{left}}_{k,b,c} \tag{3.2b}$$

$$\sigma_{\pi,s}(\mathsf{lpoint}^{B_{k,b}}_{B_{k',c}}) := \bot \quad \text{if } k' \neq k+1 \tag{3.2c}$$

$$\sigma_{\pi,s}(\mathsf{rpoint}^{B_{k,b}}_{B_{k+1,c}}) := P^{\text{right}}_{k,b,c} \tag{3.2d}$$

$$\sigma_{\pi,s}(\mathsf{rpoint}^{B_{k,b}}_{B_{k',c}}) := \bot \quad \text{if } k' \neq k+1 \tag{3.2e}$$

$$\sigma_{\pi,s}(\mathsf{lit}^{B_{k,b}}_{\ell}) := L_{k,b,\ell} \tag{3.2f}$$

$$\sigma_{\pi,s}(\mathsf{weak}^{B_{k,b}}_{A}) := W_{k,b,A} \tag{3.2g}$$

$$\sigma_{\pi,s}(\mathsf{res}^{B_{k,b}}_{x_i}) := R_{k,b,i} \tag{3.2h}$$

$$\sigma_{\pi,s}(\mathsf{derived}^{B_{k,b}}) := \bigvee_{i\in[n]} R_{k,b,i} \tag{3.2i}$$

To define the substitution on the distinguished block $B_{0,0}$ reserved for the final empty clause, let $C_\emptyset$ denote the final empty clause of the Resolution refutation π, and assume that $C_\emptyset$ is not included in the layers $\pi_1, \ldots, \pi_D$. Let $C^{\text{left}}_\emptyset := \text{left}_\pi(C_\emptyset)$ and $C^{\text{right}}_\emptyset := \text{right}_\pi(C_\emptyset)$ be the left and right premises of the final inference leading to $\bot$, and let $x_{i_\emptyset}$ be the variable resolved over in this final inference. Then, we define

$$\sigma_{\pi,s}(\mathsf{enable}^{B_{0,0}}) := \top \tag{3.3a}$$

$$\sigma_{\pi,s}(\mathsf{derived}^{B_{0,0}}) := \top \tag{3.3b}$$

$$\sigma_{\pi,s}(\mathsf{lit}^{B_{0,0}}_{\ell}) := \bot \quad \text{for all } \ell \in \text{Lit}_n, \tag{3.3c}$$

$$\sigma_{\pi,s}(\mathsf{weak}^{B_{0,0}}_{A}) := \bot \quad \text{for all } A \in [m] \tag{3.3d}$$

$$\sigma_{\pi,s}(\mathsf{res}^{B_{0,0}}_{x_i}) := \begin{cases} \top & \text{if } i = i_\emptyset, \\ \bot & \text{otherwise,} \end{cases} \tag{3.3e}$$

$$\sigma_{\pi,s}(\mathsf{lpoint}^{B_{0,0}}_{B_{1,c}}) := p_{C^{\text{left}}_\emptyset,c} \tag{3.3f}$$

$$\sigma_{\pi,s}(\mathsf{rpoint}^{B_{0,0}}_{B_{1,c}}) := p_{C^{\text{right}}_\emptyset,c} \tag{3.3g}$$

All other pointer variables out of the root block are sent to $\bot$.

Finally, if the flooring operation on W leaves any unused blocks, then we simply disable them. That is, for every block B not of the form $B_{0,0}$ or $B_{k,b}$ with $k \in [D]$ and $b \in [W]$, we set $\sigma_{\pi,s}(\mathsf{enable}^{B}) := \bot$. The remaining variables in B can be set to $\bot$, or any other value.

The following lemma shows that, starting from a pigeonhole principle instance, we can use the substitution $\sigma_{\pi,s}$ to derive a substituted instance of the Ref formula.

Lemma 3.1 (Correctness of the reduction within PK_2)**.** *Let φ be a CNF formula over n variables and m clauses that admits a layered Resolution refutation π in length L and depth D. For every $s \in \mathbb{N}$ such that $D \leq s < L$, there is a PK_2-derivation of the form*

$$\text{PHP}^L_W \vdash_{\mathsf{PK}_2} \text{Ref}_s(\varphi)\restriction_{\sigma_{\pi,s}}$$

in total size $\text{poly}(L, D, s, n, m)$*, where $\sigma_{\pi,s}$ is the depth-2 substitution described above and $W = \lfloor s/D \rfloor$.*

Proof. Starting from the clauses of PHP^L_W, we show how to derive every instance of the clauses of type (Ref-1) to (Ref-11) under the substitution $\sigma_{\pi,s}$.

The clauses of type (Ref-1) and (Ref-2). We carry out the derivation for axioms of type (Ref-1); the case (Ref-2) is dual. Furthermore, we focus on the only non-trivial cases, when the pointers are for consecutive layers k and $k+1$. Then, an axiom of type (Ref-1), written as a cedent, is

$$\neg\mathsf{enable}^{B_{k,b}}, \neg\mathsf{res}_{x_i}^{B_{k,b}}, \neg\mathsf{lpoint}_{B_{k+1,c}}^{B_{k,b}}, \neg\mathsf{lit}_{\ell}^{B_{k+1,c}}, \mathsf{lit}_{\ell}^{B_{k,b}} \tag{3.4}$$

where ℓ is a literal over a variable other than x_i. After substitution, this is precisely

$$\neg E_{k,b}, \neg R_{k,b,i}, \neg P^{\text{left}}_{k,b,c}, \neg L_{k+1,c,\ell}, L_{k,b,\ell}. \tag{3.5}$$

For $C \in \pi_k$, let us use the shorthand $T^{\text{left}}_{C,b,c} \coloneqq p_{C,b} \wedge p_{\text{left}_\pi(C),c}$. We can now rewrite $P^{\text{left}}_{k,b,c} = \bigvee_{C\in\pi_k} T^{\text{left}}_{C,b,c}$, and $\neg P^{\text{left}}_{k,b,c} = \bigwedge_{C\in\pi_k} \neg T^{\text{left}}_{C,b,c}$.

To carry out the derivation of (3.5), we first prove the following local cedent. Fix $C, C' \in \pi_k$, $D \in \pi_{k+1}$, where C' resolves over x_i and $\ell \in D$. We claim that PHP^L_W has a constant-size PK_2-derivation of

$$\neg T^{\text{left}}_{C,b,c}, \neg p_{C',b}, \neg p_{D,c}, L_{k,b,\ell}. \tag{3.6}$$

We distinguish three cases.

Case $C \neq C'$. In the first case, suppose that $C \neq C'$. Then the injectivity axiom of PHP^L_W at hole b gives the cedent $\neg p_{C,b}, \neg p_{C',b}$. By weakening, we add the formulas $\neg p_{\text{left}_\pi(C),c}$, $\neg p_{D,c}$, and $L_{k,b,\ell}$, obtaining

$$\neg p_{C,b}, \neg p_{\text{left}_\pi(C),c}, \neg p_{C',b}, \neg p_{D,c}, L_{k,b,\ell}. \tag{3.7}$$

We now apply $\vee$-introduction to derive

$$\neg p_{C,b} \vee \neg p_{\text{left}_\pi(C),c}, \neg p_{C',b}, \neg p_{D,c}, L_{k,b,\ell}. \tag{3.8}$$

Since

$$\neg T^{\text{left}}_{C,b,c} = \neg\big(p_{C,b} \wedge p_{\text{left}_\pi(C),c}\big) = \neg p_{C,b} \vee \neg p_{\text{left}_\pi(C),c}, \tag{3.9}$$

we have precisely derived eq. (3.6).

Case $C = C'$ but $D \neq \text{left}_\pi(C)$. In this case, the injectivity axiom (PHP-2) at hole c gives $\neg p_{\text{left}_\pi(C),c}, \neg p_{D,c}$. After weakening by $\neg p_{C,b}$, $\neg p_{C',b}$, and $L_{k,b,\ell}$, we again apply $\vee$-introduction to obtain (3.6).

Case $C = C'$ and $D = \text{left}_\pi(C)$. Since $\ell \in D = \text{left}_\pi(C)$ and ℓ is not over the pivot x_i, correctness of the Resolution inference implies $\ell \in C$. This uses the fact that π is a correct Resolution refutation. As a result, $p_{C,b}$ is a disjunct of $L_{k,b,\ell}$. Therefore, from the axiom $\neg p_{C,b}, p_{C,b}$, we derive by weakening and $\vee$-introduction

$$\neg p_{C,b}, L_{k,b,\ell}. \tag{3.10}$$

Weakening by $\neg p_{\text{left}_\pi(C),c}$, $\neg p_{C',b}$, and $\neg p_{D,c}$, and then applying $\vee$-introduction to both $\neg p_{C,b}$ and $\neg p_{\text{left}_\pi(C),c}$, gives (3.6).

We now collect these local cedents. Fix $C, C' \in \pi_k$ with C' resolving over x_i. Combining (3.6)

by repeated $\wedge$-introduction over all $D \in \pi_{k+1}$ with $\ell \in D$, we obtain

$$\neg T^{\text{left}}_{C,b,c}, \neg p_{C',b}, \bigwedge_{\substack{D \in \pi_{k+1} \\ \ell \in D}} \neg p_{D,c}, L_{k,b,\ell}. \tag{3.11}$$

By the definition of $L_{k+1,c,\ell}$, this is

$$\neg T^{\text{left}}_{C,b,c}, \neg p_{C',b}, \neg L_{k+1,c,\ell}, L_{k,b,\ell}. \tag{3.12}$$

Next, fixing $C \in \pi_k$ and combining over all $C' \in \pi_k$ that resolve over x_i, we derive

$$\neg T^{\text{left}}_{C,b,c}, \neg R_{k,b,i}, \neg L_{k+1,c,\ell}, L_{k,b,\ell}. \tag{3.13}$$

Finally, combining over all $C \in \pi_k$ gives

$$\neg P^{\text{left}}_{k,b,c}, \neg R_{k,b,i}, \neg L_{k+1,c,\ell}, L_{k,b,\ell}. \tag{3.14}$$

Weakening by $\neg E_{k,b}$ yields the desired cedent in eq. (3.5). This completes the derivation of the clauses of type (Ref-1). Only injectivity clauses of PHP^L_W are used.

The clauses of type (Ref-3). Written in cedent form, such axioms are of the form

$$\neg \mathsf{enable}^{B_{k,b}}, \neg \mathsf{weak}^{B_{k,b}}_A, \neg \mathsf{a\text{-}lit}^A_\ell, \mathsf{lit}^{B_{k,b}}_\ell, \tag{3.15}$$

except here $\mathsf{a\text{-}lit}^A_\ell$ is already a constant: it is $\top$ if ℓ appears in the initial clause A of φ, and $\bot$ otherwise. The only interesting case is when $\ell \in A$. After substitution, we want to derive

$$\neg E_{k,b}, \neg W_{k,b,A}, L_{k,b,\ell}. \tag{3.16}$$

In fact, we will derive the stronger cedent $\neg W_{k,b,A}, L_{k,b,\ell}$.

Fix $C \in \pi_k$ such that C weakens A. Since $\ell \in A$ and C weakens A, we have $\ell \in C$. Hence $p_{C,b}$ is a disjunct of $L_{k,b,\ell}$. Starting from the logical axiom $\neg p_{C,b}, p_{C,b}$ we derive, by weakening and $\vee$-introduction,

$$\neg p_{C,b}, L_{k,b,\ell}. \tag{3.17}$$

Now collect (3.17) over all $C \in \pi_k$ such that C weakens A. By repeated $\wedge$-introduction, we obtain

$$\bigwedge_{\substack{C \in \pi_k \\ C \text{ weakens } A}} \neg p_{C,b}, L_{k,b,\ell}. \tag{3.18}$$

By the definition of $W_{k,b,A}$, this is precisely $\neg W_{k,b,A}, L_{k,b,\ell}$. Finally, weakening by $\neg E_{k,b}$ yields (3.16).

The clauses of type (Ref-4). Written as a cedent, such an axiom is

$$\neg \mathsf{enable}^{B_{k,b}}, \neg \mathsf{derived}^{B_{k,b}}, \bigvee_{i \in [n]} \mathsf{res}^{B_{k,b}}_{x_i}. \tag{3.19}$$

After substitution, this becomes

$$\neg E_{k,b}, \neg \bigvee_{i\in[n]} R_{k,b,i}, \bigvee_{i\in[n]} R_{k,b,i}, \tag{3.20}$$

which obviously has a constant-size derivation. No axioms of PHP^L_W are used in this case.

The clauses of type (Ref-5). Written as a cedent, such an axiom is

$$\neg \mathsf{enable}^{B_{k,b}}, \mathsf{derived}^{B_{k,b}}, \bigvee_{A\in[m]} \mathsf{weak}_A^{B_{k,b}}. \tag{3.21}$$

After substitution, this becomes

$$\neg E_{k,b}, \bigvee_{i\in[n]} R_{k,b,i}, \bigvee_{A\in[m]} W_{k,b,A}. \tag{3.22}$$

For readability, let $R_{k,b} \coloneqq \bigvee_{i\in[n]} R_{k,b,i}$ and $W_{k,b} \coloneqq \bigvee_{A\in[m]} W_{k,b,A}$. Thus the substituted cedent (3.22) is $\neg E_{k,b}, R_{k,b}, W_{k,b}$.

We first prove a local cedent. Fix $C \in \pi_k$. We claim that there is a constant-size PK_2-derivation of

$$\neg p_{C,b}, R_{k,b}, W_{k,b}. \tag{3.23}$$

There are two cases.

Case when C is derived in π. Suppose that C is obtained by resolving over x_i. Then $p_{C,b}$ is a disjunct of $R_{k,b,i}$, and hence also of $R_{k,b}$. Starting from excluded middle $\neg p_{C,b}, p_{C,b}$, we derive by weakening and $\vee$-introduction $\neg p_{C,b}, R_{k,b}$. Weakening by $W_{k,b}$ gives (3.23).

Case when C is not derived in π. In this case, since π is a correct Resolution refutation, the clause C weakens some initial axiom A of φ. Hence $p_{C,b}$ is a disjunct of $W_{k,b,A}$, and therefore also of $W_{k,b}$. We can now proceed in a way analogous to the case above.

We now collect the local cedents. Combining (3.23) by repeated $\wedge$-introduction over all $C \in \pi_k$, we obtain

$$\bigwedge_{C\in\pi_k} \neg p_{C,b}, R_{k,b}, W_{k,b}. \tag{3.24}$$

By the definition of $E_{k,b}$, this is precisely $\neg E_{k,b}, R_{k,b}, W_{k,b}$. This completes the derivation of the clauses of type (Ref-5). No axioms of PHP^L_W are used in this case.

The clauses of type (Ref-6) and (Ref-7). We now consider axioms of type (Ref-6); the case (Ref-7) is dual, replacing $\mathrm{left}_\pi(C)$ by $\mathrm{right}_\pi(C)$ throughout.

Written as a cedent, an axiom of type (Ref-6) is

$$\neg \mathsf{enable}^{B_{k,b}}, \neg \mathsf{derived}^{B_{k,b}}, \bigvee_{\substack{B'\in[s+1]\\ B'<B_{k,b}}} \mathsf{lpoint}_{B'}^{B_{k,b}}. \tag{3.25}$$

We focus first on the non-root blocks. Since the only non-trivial pointers under σ are those going from layer k to layer $k+1$, the substituted cedent is

$$\neg E_{k,b}, \neg R_{k,b}, \bigvee_{c\in[W]} P^{\mathrm{left}}_{k,b,c}, \tag{3.26}$$

where, as above, $R_{k,b} \coloneqq \bigvee_{i\in[n]} R_{k,b,i}$. For readability, put $Q^{\mathrm{left}}_{k,b} \coloneqq \bigvee_{c\in[W]} P^{\mathrm{left}}_{k,b,c}$. Thus (3.26) becomes

$$\neg E_{k,b}, \neg R_{k,b}, Q^{\mathrm{left}}_{k,b}. \tag{3.27}$$

We first prove a local cedent. Fix $C, C' \in \pi_k$, where C' is derived. We claim that PHP^L_W has a polynomial-size PK_2-derivation of

$$\neg p_{C,b}, \neg p_{C',b}, Q^{\mathrm{left}}_{k,b}. \tag{3.28}$$

There are two cases.

Case $C \neq C'$. In this case, the injectivity axiom (PHP-2) at hole b gives $\neg p_{C,b}, \neg p_{C',b}$. Weakening by $Q^{\mathrm{left}}_{k,b}$ gives (3.28).

Case $C = C'$. Since C' is derived and $C = C'$, the clause C has a left premise $\mathrm{left}_\pi(C)$. We use the totality axiom (PHP-1) for the pigeon $\mathrm{left}_\pi(C)$, which states $\bigvee_{c\in[W]} p_{\mathrm{left}_\pi(C),c}$. Fix $c \in [W]$. Since $p_{C,b} \wedge p_{\mathrm{left}_\pi(C),c}$ is one of the disjuncts of $P^{\mathrm{left}}_{k,b,c}$, and hence of $Q^{\mathrm{left}}_{k,b}$, the logical axioms $\neg p_{C,b}, p_{C,b}$ and $\neg p_{\mathrm{left}_\pi(C),c}, p_{\mathrm{left}_\pi(C),c}$, together with weakening and $\wedge$-introduction, give

$$\neg p_{C,b}, \neg p_{\mathrm{left}_\pi(C),c}, Q^{\mathrm{left}}_{k,b}. \tag{3.29}$$

Combining (3.29) by repeated $\wedge$-introduction over all $c \in [W]$, we obtain

$$\neg p_{C,b}, \bigwedge_{c\in[W]} \neg p_{\mathrm{left}_\pi(C),c}, Q^{\mathrm{left}}_{k,b}, \tag{3.30}$$

and pulling the negations out we get $\neg p_{C,b}, \neg \bigvee_{c\in[W]} p_{\mathrm{left}_\pi(C),c}, Q^{\mathrm{left}}_{k,b}$. Cutting this cedent with the totality axiom (PHP-1) for $\mathrm{left}_\pi(C)$ gives $\neg p_{C,b}, Q^{\mathrm{left}}_{k,b}$. This completes the subderivation.

We now collect the local cedents. Fix $C \in \pi_k$. Combining (3.28) by repeated $\wedge$-introduction over all derived $C' \in \pi_k$, we obtain

$$\neg p_{C,b}, \bigwedge_{\substack{C'\in\pi_k \\ C' \text{ is derived}}} \neg p_{C',b}, Q^{\mathrm{left}}_{k,b}. \tag{3.31}$$

By the definition of $R_{k,b}$, this is $\neg p_{C,b}, \neg R_{k,b}, Q^{\mathrm{left}}_{k,b}$. Combining over all $C \in \pi_k$ by repeated $\wedge$-introduction gives

$$\bigwedge_{C\in\pi_k} \neg p_{C,b}, \neg R_{k,b}, Q^{\mathrm{left}}_{k,b}. \tag{3.32}$$

By the definition of $E_{k,b}$, this is precisely $\neg E_{k,b}, \neg R_{k,b}, Q^{\mathrm{left}}_{k,b}$. Expanding the abbreviation $Q^{\mathrm{left}}_{k,b}$ gives the desired substituted cedent in eq. (3.26).

It remains to mention the distinguished root block $B_{0,0}$. Since we set $\sigma_{\pi,s}(\mathsf{enable}^{B_{0,0}}) = \top$ and

$\sigma_{\pi,s}(\mathsf{derived}^{B_{0,0}}) = \top$, the corresponding substituted cedent is $\bigvee_{c\in[W]} p_{C_\emptyset^{\text{left}},c}$, which is exactly the totality axiom (PHP-1) for the pigeon $C_\emptyset^{\text{left}}$.

The clauses of type (Ref-8) and (Ref-9). We now consider axioms of type (Ref-9); the case (Ref-8) is analogous. Written as a cedent, such an axiom is

$$\neg\mathsf{enable}^{B_{k,b}}, \neg\mathsf{rpoint}^{B_{k,b}}_{B_{k+1,c}}, \mathsf{enable}^{B_{k+1,c}}. \tag{3.33}$$

We focus on the non-trivial case, when the pointer is from layer k to layer $k+1$. After substitution, the goal is to derive

$$\neg E_{k,b}, \neg P^{\text{right}}_{k,b,c}, E_{k+1,c}. \tag{3.34}$$

For $C \in \pi_k$, let $T^{\text{right}}_{C,b,c} := p_{C,b} \wedge p_{\text{right}_\pi(C),c}$, so $P^{\text{right}}_{k,b,c} = \bigvee_{C\in\pi_k} T^{\text{right}}_{C,b,c}$.

We first prove a local cedent. Fix $C, C' \in \pi_k$. We claim that PHP^L_W has a constant-size PK_2-derivation of

$$\neg p_{C',b}, \neg T^{\text{right}}_{C,b,c}, E_{k+1,c}. \tag{3.35}$$

There are two cases.

Case $C \neq C'$. In this case, the injectivity axiom (PHP-2) at hole b gives $\neg p_{C,b}, \neg p_{C',b}$. Weakening by $\neg p_{\text{right}_\pi(C),c}$ and $E_{k+1,c}$, and then applying $\vee$-introduction to $\neg p_{C,b}$ and $\neg p_{\text{right}_\pi(C),c}$, gives eq. (3.35).

Case $C = C'$. In this case, $\text{right}_\pi(C) \in \pi_{k+1}$. Hence $p_{\text{right}_\pi(C),c}$ is a disjunct of $E_{k+1,c}$. From the logical axiom $\neg p_{\text{right}_\pi(C),c}, p_{\text{right}_\pi(C),c}$, we derive by weakening and $\vee$-introduction

$$\neg p_{\text{right}_\pi(C),c}, E_{k+1,c}. \tag{3.36}$$

Weakening by $\neg p_{C,b}$ and $\neg p_{C',b}$, and then applying $\vee$-introduction to $\neg p_{C,b}$ and $\neg p_{\text{right}_\pi(C),c}$, gives (3.35).

We now collect the local cedents. Fix $C \in \pi_k$. Combining (3.35) by repeated $\wedge$-introduction over all $C' \in \pi_k$, we obtain

$$\bigwedge_{C'\in\pi_k} \neg p_{C',b}, \neg T^{\text{right}}_{C,b,c}, E_{k+1,c}. \tag{3.37}$$

By the definition of $E_{k,b}$, this is $\neg E_{k,b}, \neg T^{\text{right}}_{C,b,c}, E_{k+1,c}$. Finally, combining over all $C \in \pi_k$ by repeated $\wedge$-introduction gives

$$\neg E_{k,b}, \bigwedge_{C\in\pi_k} \neg T^{\text{right}}_{C,b,c}, E_{k+1,c}. \tag{3.38}$$

Since $\bigwedge_{C\in\pi_k} \neg T^{\text{right}}_{C,b,c} = \neg P^{\text{right}}_{k,b,c}$, we derive $\neg E_{k,b}, \neg P^{\text{right}}_{k,b,c}, E_{k+1,c}$, which is the desired substituted cedent (3.34).

Note that if the pointer variable is not between consecutive layers, then $\sigma_{\pi,s}(\mathsf{rpoint}^{B_{k,b}}_{B'}) = \bot$, so the substituted cedent is immediate. The case for the root block is also immediate. This completes the derivation of the clauses of type (Ref-9). Only injectivity clauses of PHP^L_W are used in the non-trivial cases of these derivations.

The clauses of type (Ref-10) and (Ref-11). We finally consider the axioms enforcing that the root block is enabled and contains the empty clause. Recall that the distinguished root block $B_{0,0}$ is identified with the final block $s+1$ of $\mathrm{Ref}_s(\varphi)$.

It is easy to see that the axioms of type (Ref-10) are trivial to derive, since written as cedents, these are $\neg \mathsf{lit}_\ell^{B_{0,0}}$ for all $\ell \in \mathrm{Lit}_n$, and after substitution, this becomes $\neg\sigma_{\pi,s}(\mathsf{lit}_\ell^{B_{0,0}})$. This is immediate, since $\sigma_{\pi,s}(\mathsf{lit}_\ell^{B_{0,0}}) = \bot$. Similarly, for the axioms of type (Ref-11), written as a cedent, we have $\mathsf{enable}^{B_{0,0}}$, $\sigma_{\pi,s}(\mathsf{enable}^{B_{0,0}}) = \top$. No axioms of PHP_W^L are used in these last cases. □

The correctness of the reduction implies the following result for PK_d systems.

Theorem 3.2. *Let φ be an unsatisfiable CNF formula over n variables and m clauses that admits a layered Resolution refutation π in length L and depth D, and let $d \in \mathbb{N}$. For every $s \in \mathbb{N}$ such that $D \leq s < L$, if PK_d admits a refutation of $\mathrm{Ref}_s(\varphi)$ in size T, then PK_{d+2} admits a refutation of PHP_W^L in size $T \cdot \mathrm{poly}(L, D, s, n, m)$, where $W = \lfloor s/D \rfloor$.*

Proof. By Lemma 3.1, starting from the axioms of PHP_W^L, we have a derivation of the form

$$\mathrm{PHP}_W^L \vdash_{\mathsf{PK}_2} \mathrm{Ref}_s(\varphi)\restriction_{\sigma_{\pi,s}} \vdash_{\mathsf{PK}_{d+2}} \bot, \tag{3.39}$$

where the first half of the refutation follows from the correctness of the substitution $\sigma_{\pi,s}$ and the second half is obtained by substituting $\sigma_{\pi,s}$ in the PK_d-refutation of $\mathrm{Ref}_s(\varphi)$ to get a PK_{d+2}-refutation of $\mathrm{Ref}_s(\varphi)\restriction_{\sigma_{\pi,s}}$. The size blow-up is bounded by $T \cdot \mathrm{poly}(L, D, s, n, m)$. □

3.2 A framework for reductions in $\mathsf{Res}(k)$

To work within the $\mathsf{Res}(k)$ subsystems of PK we introduce a variant of the reduction above. For every variable, it maps its positive and negative literals to different non-complementary formulas. Technically speaking, such a substitution cannot preserve the validity of a refutation in general. To address this, we first prove a general lemma which argues that, in the presence of the PHP axioms, we can afford to map positive and negative literals into different formulas, to avoid the increase in depth caused by distributing negations, and preserve validity.

This is precisely the notion of *contextual substitutions* defined below.

Definition 3.3 (Contextual substitutions)**.** Let $\varphi(x_1, \ldots, x_n)$ and $\psi(y_1, \ldots, y_m)$ be CNF formulas over disjoint sets of variables, and let τ be a mapping from the literals of ψ to formulas over the variables of φ. We say that τ is a *(k, r)-substitution for ψ in the context of φ in size t* if the following conditions hold:

(i) for every literal $\ell \in \mathrm{Lit}(\psi)$, the formula $\tau(\ell)$ is an r-DNF;

(ii) for every variable y_i of ψ, there is a $\mathsf{Res}(r)$-derivation of $\tau(y_i) \vee \tau(\neg y_i)$ from φ in size at most t;

(iii) for every clause C of ψ, there is a $\mathsf{Res}(rk)$-derivation of $\tau(C)$ from φ in size at most t;

(iv) for every variable y_i of ψ, every term U of $\tau(y_i)$, and every term V of $\tau(\neg y_i)$, there is a $\mathsf{Res}(r)$-derivation of the clause $\neg U \vee \neg V$ from φ in size at most t.

We now argue that these contextual substitutions can be used to chain together refutations within $\mathsf{Res}(k)$.

Lemma 3.4 (Chaining lemma for contextual substitutions)**.** *Let $\varphi(x_1, \ldots, x_n)$ and $\psi(y_1, \ldots, y_m)$ be CNF formulas and let τ be a (k, r)-substitution for ψ in the context of φ in size t. Suppose moreover that every r-DNF in the image of τ has at most M terms. If ψ admits a $\mathsf{Res}(k)$-refutation in size S, then φ admits a $\mathsf{Res}(rk)$-refutation in size at most $S \cdot t \cdot M^{O(k)}$.*

Proof. Let π be a $\mathsf{Res}(k)$-refutation of ψ in size S. We apply τ to every literal appearing in π, and distribute conjunctions over disjunctions to obtain DNF formulas. Throughout the proof, for a DNF formula D, we write $\tau(D)$ to denote the resulting DNF formula obtained by applying τ on the literals and then distributing connectives back into a DNF again.

Since every term in a k-DNF contains at most k literals, and the image of each literal is an r-DNF with at most M terms, every translated term has width at most rk and expands into at most M^k terms. Thus, for every line D in π, the translated line $\tau(D)$ is an rk-DNF of size at most $S \cdot M^{O(k)}$.

We show by induction on the lines of π that each translated line $\tau(D)$ admits a $\mathsf{Res}(rk)$-derivation from φ within the desired size bound.

Initial clauses. The base case is when D is an initial clause of ψ. In this case, item (iii) in the definition of contextual substitutions guarantees that there is a derivation of $\tau(D)$ from φ in size at most t.

Weakening. If the inference is a weakening step, then after translation a single weakening step suffices to derive the translated line.

Excluded middle. The case when D is the logical axiom $y_i \vee \neg y_i$ is covered by item (ii), which ensures a $\mathsf{Res}(r)$-derivation of $\tau(y_i) \vee \tau(\neg y_i)$ from φ in size at most t.

$\wedge$-Introduction. Suppose the inference rule is

$$\frac{\Gamma \vee T_1 \qquad \Gamma \vee T_2}{\Gamma \vee (T_1 \wedge T_2)} \ (\wedge)\text{-intro}$$

where $T_1 \wedge T_2$ has width at most k. By induction, after applying the substitution τ, we may assume we have already derived $\tau(\Gamma) \vee \tau(T_1)$ and $\tau(\Gamma) \vee \tau(T_2)$.

Let us write $\tau(T_1) = U_1 \vee \cdots \vee U_a$ and $\tau(T_2) = V_1 \vee \cdots \vee V_b$, so that the two available translated lines are $\tau(\Gamma) \vee U_1 \vee \cdots \vee U_a$ and $\tau(\Gamma) \vee V_1 \vee \cdots \vee V_b$. We need to derive

$$\tau(\Gamma) \vee \bigvee_{\alpha \in [a],\, \beta \in [b]} (U_\alpha \wedge V_\beta). \tag{3.40}$$

We do it as follows. Let $P_0 \coloneqq \tau(\Gamma) \vee U_1 \vee \cdots \vee U_a$, which is an available premise, and for $\alpha \in [a]$, let

$$P_\alpha \coloneqq \tau(\Gamma) \vee \bigvee_{\alpha' \leq \alpha,\, \beta \in [b]} (U_{\alpha'} \wedge V_\beta) \vee U_{\alpha+1} \vee \cdots \vee U_a, \tag{3.41}$$

so that P_a is the desired line (3.40).

Assume $P_{\alpha-1}$ has been derived. Then $P_{\alpha-1}$ has the form $\Delta_\alpha \vee U_\alpha$, where

$$\Delta_\alpha \coloneqq \tau(\Gamma) \vee \bigvee_{\alpha' < \alpha,\, \beta \in [b]} (U_{\alpha'} \wedge V_\beta) \vee U_{\alpha+1} \vee \cdots \vee U_a. \tag{3.42}$$

From the second translated premise $\tau(\Gamma) \vee V_1 \vee \cdots \vee V_b$ we can use weakening to derive

$$\Delta_\alpha \vee V_1 \vee \cdots \vee V_b. \tag{3.43}$$

It now suffices to apply the product rule (Lemma 2.1) on $P_{\alpha-1}$ and (3.43) to obtain

$$\Delta_\alpha \vee (U_\alpha \wedge V_1) \vee \cdots \vee (U_\alpha \wedge V_b), \tag{3.44}$$

which is exactly P_α.

Note that this preserves the bound on the width of the terms: every product term has width at most rk. The construction uses $O(ab)$ inference steps, and every intermediate formula has at most $M^{O(k)}$ times as many terms as the original premises. Since $a, b \leq M^k$, the total size of the derivation, including the encoding of all intermediate formulas, is bounded by the size of the original premises times $M^{O(k)}$.

The cut rule. Suppose the inference is a cut on the term $T = \ell_1 \wedge \cdots \wedge \ell_s$, with $s \leq k$, of the form

$$\frac{\Gamma \vee T \qquad \Delta \vee \neg\ell_1 \vee \cdots \vee \neg\ell_s}{\Gamma \vee \Delta}\ \text{Cut}$$

where, by induction, we have already derived $\tau(\Gamma) \vee \tau(T)$ and $\tau(\Delta) \vee \tau(\neg\ell_1) \vee \cdots \vee \tau(\neg\ell_s)$.

Suppose the translation of T consists of a disjuncts $\tau(T) = U_1 \vee \cdots \vee U_a$. Observe that each U_α for $\alpha \in [a]$ is obtained by choosing, for each $j \in [s]$, one term $U_{\alpha,j}$ of $\tau(\ell_j)$ such that $U_\alpha = U_{\alpha,1} \wedge \cdots \wedge U_{\alpha,s}$. Let us also say that the translation of $\neg T$ consists of b disjuncts $\tau(\neg\ell_1) \vee \cdots \vee \tau(\neg\ell_s) = V_1 \vee \cdots \vee V_b$. Note that each V_β for $\beta \in [b]$ is a term of $\tau(\neg\ell_j)$ for some $j \in [s]$.

We first show how to handle a fixed term U_α. For each V_β, choose $j = j(\beta)$ such that V_β is a term of $\tau(\neg\ell_j)$. Since $U_{\alpha,j}$ is a term of $\tau(\ell_j)$, item (iv) gives a derivation from φ of $\neg U_{\alpha,j} \vee \neg V_\beta$. By weakening, this gives

$$\neg U_\alpha \vee \neg V_\beta, \tag{3.45}$$

because U_α contains $U_{\alpha,j}$ as a subterm.

We now derive $\tau(\Delta) \vee \neg U_\alpha$ from $\tau(\Delta) \vee V_1 \vee \cdots \vee V_b$, which is one of the premises given by induction hypothesis. Starting with this line, eliminate the terms $V_1, \ldots, V_b$ one at a time. Denote

$$R_\beta \coloneqq \tau(\Delta) \vee V_\beta \vee V_{\beta+1} \vee \cdots \vee V_b \vee \neg U_\alpha. \tag{3.46}$$

For $\beta = 1$, we can weaken the premise $\tau(\Delta) \vee V_1 \vee \cdots \vee V_b$ with $\neg U_\alpha$ to get R_1. If R_β has already been derived, then we can cut with the local incompatibility derivation of $\neg U_\alpha \vee \neg V_\beta$ in (3.45) to get

$$\tau(\Delta) \vee V_{\beta+1} \vee \cdots \vee V_b \vee \neg U_\alpha. \tag{3.47}$$

This eventually yields, for every $\alpha \in [a]$, the line

$$\tau(\Delta) \vee \neg U_\alpha. \tag{3.48}$$

At this point it suffices to successively cut the a lines of the form (3.48) with the translated premise $\tau(\Gamma) \vee \tau(T) = \tau(\Gamma) \vee U_1 \vee \cdots \vee U_a$ to obtain $\tau(\Gamma) \vee \tau(\Delta)$, which is precisely the translation of the conclusion.

This completes the induction over the lines of π. In each case, the number of inference steps and the size of the intermediate formulas are bounded by $t \cdot M^{O(k)}$ times the size of the corresponding original lines. Summing over the lines of π, whose total size is S, we obtain a $\mathrm{Res}(rk)$-refutation of φ in size at most $S \cdot t \cdot M^{O(k)}$. □

3.3 The reduction in Res(k)

We are ready to define the variant of the reduction $\sigma_{\pi,s}$ described in Section 3.1. The setup is the same as before: we fix φ to be an unsatisfiable CNF formula over n variables and m clauses, and π to be a layered Resolution refutation of φ consisting of L clauses and depth D. The idea is to reuse the same substitution as before, but this time assigning different DNF formulas to positive and negative literals in a complementary fashion. We denote this new substitution by $\tau_{\pi,s}$, where $s \in \mathbb{N}$ is again the size parameter of the $\textsc{Ref}_s(\varphi)$ formula we substitute into, and we again set $W := \lfloor s/D \rfloor$.

The substitution $\tau_{\pi,s}$ agrees with $\sigma_{\pi,s}$ on all positive literals. Thus, for every variable z of $\textsc{Ref}_s(\varphi)$, we set $\tau_{\pi,s}(z) := \sigma_{\pi,s}(z)$. It remains to define the images of the negated literals.

For convenience, for every $k \in [D]$ and $b \in [W]$, let

$$\overline{E}_{k,b} := \bigvee_{C \in \pi \setminus \pi_k} p_{C,b}, \tag{3.49}$$

and notice that $E_{k,b} \vee \overline{E}_{k,b} = \bigvee_{C \in \pi} p_{C,b}$.

For non-root blocks $B_{k,b}$, where $k \in [D]$ and $b \in [W]$, we define the following DNF formulas:

$$\overline{L}_{k,b,\ell} := \bigvee_{\substack{C \in \pi \\ C \notin \pi_k \text{ or } \ell \notin C}} p_{C,b} \tag{3.50a}$$

$$\overline{W}_{k,b,A} := \bigvee_{\substack{C \in \pi \\ C \notin \pi_k \text{ or not weakening } A}} p_{C,b}, \tag{3.50b}$$

$$\overline{R}_{k,b,i} := \bigvee_{\substack{C \in \pi \\ C \notin \pi_k \text{ or not resolving } x_i}} p_{C,b} \tag{3.50c}$$

$$\overline{D}_{k,b} := \bigvee_{\substack{C \in \pi \\ C \notin \pi_k \text{ or not derived}}} p_{C,b}. \tag{3.50d}$$

We next define the negative images of the pointer variables. Recall that, in the previous substitution, for consecutive layers we have $P^{\text{left}}_{k,b,c} = \bigvee_{C \in \pi_k} \big(p_{C,b} \wedge p_{\text{left}_\pi(C),c}\big)$, and analogously for right pointers. For $k < D$, we define the "negation" of those as their complementary disjunctions,

$$\overline{P}^{\text{left}}_{k,b,c} := \bigvee_{\substack{C,D \in \pi \\ C \notin \pi_k \text{ or } D \neq \text{left}_\pi(C)}} \big(p_{C,b} \wedge p_{D,c}\big), \tag{3.51a}$$

$$\overline{P}^{\text{right}}_{k,b,c} := \bigvee_{\substack{C,D \in \pi \\ C \notin \pi_k \text{ or } D \neq \text{right}_\pi(C)}} \big(p_{C,b} \wedge p_{D,c}\big). \tag{3.51b}$$

Consequently, $P^{\text{left}}_{k,b,c} \vee \overline{P}^{\text{left}}_{k,b,c} = \bigvee_{C,D \in \pi} \big(p_{C,b} \wedge p_{D,c}\big)$, and analogously for right pointers.

The substitution $\tau_{\pi,s}$ on negated literals of non-root blocks is now defined as follows:

$$\tau_{\pi,s}(\neg \mathsf{enable}^{B_{k,b}}) := \overline{E}_{k,b} \tag{3.52a}$$

$$\tau_{\pi,s}(\neg \mathsf{lpoint}^{B_{k,b}}_{B_{k+1,c}}) := \overline{P}^{\text{left}}_{k,b,c} \tag{3.52b}$$

$$\tau_{\pi,s}(\neg \mathsf{lpoint}^{B_{k,b}}_{B_{k',c}}) := \top \quad \text{if } k' \neq k+1 \tag{3.52c}$$

$$\tau_{\pi,s}(\neg \mathsf{rpoint}^{B_{k,b}}_{B_{k+1,c}}) := \overline{P}^{\text{right}}_{k,b,c} \tag{3.52d}$$

$$\tau_{\pi,s}(\neg \mathsf{rpoint}^{B_{k,b}}_{B_{k',c}}) := \top \quad \text{if } k' \neq k+1 \tag{3.52e}$$

$$\tau_{\pi,s}(\neg \mathsf{lit}^{B_{k,b}}_{\ell}) := \overline{L}_{k,b,\ell} \tag{3.52f}$$

$$\tau_{\pi,s}(\neg \mathsf{weak}^{B_{k,b}}_{A}) := \overline{W}_{k,b,A} \tag{3.52g}$$

$$\tau_{\pi,s}(\neg \mathsf{res}^{B_{k,b}}_{x_i}) := \overline{R}_{k,b,i} \tag{3.52h}$$

$$\tau_{\pi,s}(\neg \mathsf{derived}^{B_{k,b}}) := \overline{D}_{k,b}. \tag{3.52i}$$

It remains to define $\tau_{\pi,s}$ on negated literals of the distinguished root block $B_{0,0}$, which is identified with the final block $s+1$. Recall that $C^{\text{left}}_{\emptyset}$ and $C^{\text{right}}_{\emptyset}$ are the left and right premises of the final inference, and that $x_{i_\emptyset}$ is the variable resolved over in that inference. We set

$$\tau_{\pi,s}(\neg \mathsf{enable}^{B_{0,0}}) := \bot \tag{3.53a}$$

$$\tau_{\pi,s}(\neg \mathsf{derived}^{B_{0,0}}) := \bot \tag{3.53b}$$

$$\tau_{\pi,s}(\neg \mathsf{weak}_A^{B_{0,0}}) := \top \quad \text{for all } A \in [m] \tag{3.53d}$$

$$\tau_{\pi,s}(\neg \mathsf{res}_{x_i}^{B_{0,0}}) := \begin{cases} \bot & \text{if } i = i_\emptyset, \\ \top & \text{otherwise,} \end{cases} \tag{3.53e}$$

$$\tau_{\pi,s}(\neg \mathsf{lit}_\ell^{B_{0,0}}) := \top \quad \text{for all } \ell \in \mathrm{Lit}_n \tag{3.53c}$$

$$\tau_{\pi,s}(\neg \mathsf{lpoint}_{B_{1,c}}^{B_{0,0}}) := \bigvee_{\substack{D \in \pi \\ D \neq C_\emptyset^{\mathrm{left}}}} p_{D,c} \tag{3.53f}$$

$$\tau_{\pi,s}(\neg \mathsf{rpoint}_{B_{1,c}}^{B_{0,0}}) := \bigvee_{\substack{D \in \pi \\ D \neq C_\emptyset^{\mathrm{right}}}} p_{D,c}. \tag{3.53g}$$

All other negated pointer variables out of the root block are sent to $\top$.

Finally, if the flooring operation on W leaves unused blocks among the first s blocks, then we disable them. That is, for every block B not of the form $B_{0,0}$ or $B_{k,b}$ with $k \in [D]$ and $b \in [W]$, we set $\tau_{\pi,s}(\mathsf{enable}^B) := \bot$ and $\tau_{\pi,s}(\neg \mathsf{enable}^B) := \top$. All other positive literals belonging to such a block are sent to $\bot$, and their negations are sent to $\top$.

We now prove that $\tau_{\pi,s}$ is a correct contextual substitution for $\mathsf{Res}(k)$ in the context of the onto pigeonhole principle.

Lemma 3.5 (Correctness of the reduction)**.** *Let φ be a CNF formula over n variables and m clauses that admits a layered Resolution refutation π of length L and depth D. For every $s \in \mathbb{N}$ such that $D \leq s < L$, the substitution $\tau_{\pi,s}$ defined above is a $(k, 2)$-substitution for $\textsc{Ref}_s(\varphi)$ in the context of $\textsc{ontoPHP}_W^L$, in size $\mathrm{poly}(L, D, s, n, m)$, where $W = \lfloor s/D \rfloor$.*

Proof. We verify the four conditions in the definition of contextual substitutions (Definition 3.3).

First, every image of a literal under $\tau_{\pi,s}$ is a 2-DNF. Indeed, on positive literals the substitution agrees with $\sigma_{\pi,s}$, the formulas $\overline{E}, \overline{L}, \overline{W}, \overline{R}, \overline{D}$ are all positive clauses, and the formulas for negated pointer variables are 2-DNF formulas.

We next verify the excluded-middle condition in item (ii). For the enable variables we have

$$\tau_{\pi,s}(\mathsf{enable}^{B_{k,b}}) \vee \tau_{\pi,s}(\neg \mathsf{enable}^{B_{k,b}}) = E_{k,b} \vee \overline{E}_{k,b} = \bigvee_{C \in \pi} p_{C,b}, \tag{3.54}$$

which is precisely the onto axiom (PHP-3) at hole b.

Similarly, for the variables $\mathsf{lit}_\ell^{B_{k,b}}$, we have

$$\tau_{\pi,s}(\mathsf{lit}_\ell^{B_{k,b}}) \vee \tau_{\pi,s}(\neg \mathsf{lit}_\ell^{B_{k,b}}) = L_{k,b,\ell} \vee \overline{L}_{k,b,\ell} = \bigvee_{C \in \pi} p_{C,b}. \tag{3.55}$$

The same argument applies to the variables $\mathsf{weak}_A^{B_{k,b}}$, $\mathsf{res}_{x_i}^{B_{k,b}}$, and $\mathsf{derived}^{B_{k,b}}$, with the disjunction being the onto axiom at hole b.

For pointer variables between consecutive layers, we treat the case for left pointers; the case for right pointers is dual. By (3.51a),

$$\tau_{\pi,s}(\mathsf{lpoint}_{B_{k+1,c}}^{B_{k,b}}) \vee \tau_{\pi,s}(\neg \mathsf{lpoint}_{B_{k+1,c}}^{B_{k,b}}) = \bigvee_{C,D \in \pi} \left(p_{C,b} \wedge p_{D,c}\right). \tag{3.56}$$

We show that (3.56) has a polynomial-size $\mathsf{Res}(2)$-derivation from the onto axioms. We distinguish two cases.

Case $b \neq c$. We derive (3.56) by repeated applications of the product rule (Lemma 2.1). Enumerate the clauses of π as $\pi = \{C_1, \ldots, C_L\}$, and for every $i \in \{0, \ldots, L\}$, consider the formula

$$F_i := \bigvee_{h=1}^{i} \bigvee_{D \in \pi} (p_{C_h,b} \wedge p_{D,c}) \;\vee\; \bigvee_{h=i+1}^{L} p_{C_h,b}. \tag{3.57}$$

The formula F_0 is precisely the onto axiom at hole b. Suppose that F_{i-1} has been derived. By weakening the onto axiom at hole c, we obtain

$$\bigvee_{h=1}^{i-1} \bigvee_{D \in \pi} (p_{C_h,b} \wedge p_{D,c}) \;\vee\; \bigvee_{h=i+1}^{L} p_{C_h,b} \;\vee\; \bigvee_{D \in \pi} p_{D,c}. \tag{3.58}$$

Applying the product rule to this formula and F_{i-1}, with the common context consisting of all terms except $p_{C_i,b}$, yields F_i. Each application of the product rule can be simulated in Res(2) using $O(L)$ inference steps. After L iterations, we obtain $F_L = \bigvee_{C,D \in \pi}(p_{C,b} \wedge p_{D,c})$, which is exactly (3.56).

Case $b = c$. In this case, we start from the onto axiom at hole b and for each clause $C \in \pi$, we use the product rule with the single term $p_{C,b}$ to replace the corresponding disjunct by $p_{C,b} \wedge p_{C,b}$. This gives $\bigvee_{C \in \pi}(p_{C,b} \wedge p_{C,b})$. By weakening with all the remaining terms, we obtain $\bigvee_{C,D \in \pi}(p_{C,b} \wedge p_{D,b})$, which is again (3.56).

Now, if the pointer is not between consecutive layers, then its positive image is $\bot$ and its negative image is $\top$, so excluded middle follows. The case for disabled blocks follows immediately, and for the non-trivial root pointers we have

$$\tau_{\pi,s}(\mathsf{lpoint}_{B_{1,c}}^{B_{0,0}}) \vee \tau_{\pi,s}(\neg \mathsf{lpoint}_{B_{1,c}}^{B_{0,0}}) = p_{C_\emptyset^{\mathrm{left}},c} \vee \bigvee_{\substack{D \in \pi \\ D \neq C_\emptyset^{\mathrm{left}}}} p_{D,c}, \tag{3.59}$$

which is exactly the onto axiom at c.

We now verify condition (iv). For the enable variables, take a term $p_{C,b}$ of $E_{k,b}$ and a term $p_{C',b}$ of $\overline{E}_{k,b}$. Then $C \in \pi_k$ and $C' \notin \pi_k$, so $C \neq C'$. Hence the injectivity axiom (PHP-2) gives $\neg p_{C,b} \vee \neg p_{C',b}$. This is precisely the required incompatibility clause.

The same argument applies to $\mathsf{lit}_\ell^{B_{k,b}}$, $\mathsf{weak}_A^{B_{k,b}}$, $\mathsf{res}_{x_i}^{B_{k,b}}$, and $\mathsf{derived}^{B_{k,b}}$, since a term from the positive image and a term from the negative image are complementary at the same hole b, and therefore follow from the injectivity axioms.

For pointer variables, consider again the case of left pointers between consecutive layers. A term of the positive image has the form $p_{C,b} \wedge p_{\mathrm{left}_\pi(C),c}$, where $C \in \pi_k$, and a term of the negative image has the form $p_{C',b} \wedge p_{D,c}$, where either $C' \notin \pi_k$ or $D \neq \mathrm{left}_\pi(C')$. Consequently, $(C, \mathrm{left}_\pi(C)) \neq (C', D)$. If $C \neq C'$, the injectivity axiom at hole b gives $\neg p_{C,b} \vee \neg p_{C',b}$, and by weakening this yields the desired clause

$$\neg\big(p_{C,b} \wedge p_{\mathrm{left}_\pi(C),c}\big) \vee \neg\big(p_{C',b} \wedge p_{D,c}\big). \tag{3.60}$$

If $C = C'$, then necessarily $D \neq \mathrm{left}_\pi(C)$, and the injectivity axiom at hole c gives $\neg p_{\mathrm{left}_\pi(C),c} \vee \neg p_{D,c}$. Again, weakening gives the desired clause. The case for right-side pointers is dual. Root pointers are handled in the same way using injectivity at hole c, and all constant cases are immediate or vacuous.

It remains to check (iii): every initial clause of $\mathrm{Ref}_s(\varphi)$, after applying $\tau_{\pi,s}$, has a polynomial-size

Res($2k$)-derivation from $\textsc{ontophp}^L_W$. We go through the different kinds of axioms.

The clauses of type (Ref-1) and (Ref-2). We treat the left case; the right case is dual. For a non-root block and a consecutive left pointer, the translated cedent is

$$\overline{E}_{k,b}, \overline{R}_{k,b,i}, \overline{P}^{\text{left}}_{k,b,c}, \overline{L}_{k+1,c,\ell}, L_{k,b,\ell}, \tag{3.61}$$

where ℓ is a literal over a variable different from the pivot x_i. From the onto axioms at holes b and c, by the product rule, we derive

$$\bigvee_{C,D\in\pi} (p_{C,b} \wedge p_{D,c}). \tag{3.62}$$

As above, when $b = c$, the same formula follows by retaining the diagonal terms of the onto axiom and then weakening.

Fix a pair of clauses $C, D \in \pi$. We show that the term $p_{C,b} \wedge p_{D,c}$ can be weakened to one of the terms in (3.61). If $C \notin \pi_k$, then $p_{C,b}$ is a disjunct of $\overline{E}_{k,b}$. Otherwise, if C does not resolve over x_i, then $p_{C,b}$ is a disjunct of $\overline{R}_{k,b,i}$. If $D \neq \text{left}_\pi(C)$, then $p_{C,b} \wedge p_{D,c}$ is a disjunct of $\overline{P}^{\text{left}}_{k,b,c}$. If $D \notin \pi_{k+1}$ or $\ell \notin D$, then $p_{D,c}$ is a disjunct of $\overline{L}_{k+1,c,\ell}$. Finally, if none of these cases holds, then $C \in \pi_k$ resolves over x_i, $D = \text{left}_\pi(C) \in \pi_{k+1}$, and $\ell \in D$. Since ℓ is not over the pivot x_i, correctness of the Resolution inference implies $\ell \in C$, and hence $p_{C,b}$ is a disjunct of $L_{k,b,\ell}$. It follows that each term in (3.62) can be weakened to a term of (3.61).

If the pointer is not between consecutive layers, then $\tau_{\pi,s}(\neg\mathsf{lpoint}^{B_{k,b}}_{B'}) = \top$, and the translated axiom is immediate. The corresponding root cases are also immediate unless the pointer is one of the distinguished root pointers. In that case the same argument applies using the onto axiom at c and the correctness of the final inference deriving the empty clause.

The clauses of type (Ref-3). If $\ell \notin A$, then $\tau_{\pi,s}(\neg\mathsf{a\text{-}lit}^A_\ell) = \top$, and the translated axiom is immediate. Suppose therefore that $\ell \in A$. The translated cedent is

$$\overline{E}_{k,b}, \overline{W}_{k,b,A}, L_{k,b,\ell}. \tag{3.63}$$

We derive it from the onto axiom $\bigvee_{C\in\pi} p_{C,b}$. Fix $C \in \pi$. If $C \notin \pi_k$, then $p_{C,b}$ is a disjunct of $\overline{E}_{k,b}$. If $C \in \pi_k$ but C does not weaken A, then $p_{C,b}$ is a disjunct of $\overline{W}_{k,b,A}$. Finally, if $C \in \pi_k$ weakens A, then, since $\ell \in A$, we have $\ell \in C$, and $p_{C,b}$ is a disjunct of $L_{k,b,\ell}$. Hence every term of the onto axiom weakens to a term of (3.63).

The clauses of type (Ref-4). The translated cedent is

$$\overline{E}_{k,b}, \overline{D}_{k,b}, \bigvee_{i\in[n]} R_{k,b,i}. \tag{3.64}$$

This follows directly from the onto axiom at b. Indeed, for every $C \in \pi$, if $C \notin \pi_k$, then $p_{C,b}$ is a disjunct of $\overline{E}_{k,b}$. If $C \in \pi_k$ is not derived, then it is a disjunct of $\overline{D}_{k,b}$. Finally, if $C \in \pi_k$ is derived, then it resolves over some x_i, and $p_{C,b}$ is a disjunct of $R_{k,b,i}$.

The clauses of type (Ref-5). The translated cedent is

$$\overline{E}_{k,b}, \bigvee_{i\in[n]} R_{k,b,i}, \bigvee_{A\in[m]} W_{k,b,A}. \tag{3.65}$$

Again, this follows from the onto axiom at b. If $C \notin \pi_k$, then $p_{C,b}$ is a disjunct of $\overline{E}_{k,b}$. If $C \in \pi_k$ is derived, then it is a disjunct of $R_{k,b,i}$ for the variable x_i resolved over. If $C \in \pi_k$ is not derived, then C weakens some initial clause A of φ, and $p_{C,b}$ is a disjunct of $W_{k,b,A}$.

The clauses of type (Ref-6) and (Ref-7). We consider the left-pointer case; the right-pointer case is dual. The translated cedent for a non-root block is

$$\overline{E}_{k,b}, \overline{D}_{k,b}, \bigvee_{c\in[W]} P^{\text{left}}_{k,b,c}. \tag{3.66}$$

We derive this formula from the onto axiom at b. Fix $C \in \pi$. If $C \notin \pi_k$, then $p_{C,b}$ is a disjunct of $\overline{E}_{k,b}$. If $C \in \pi_k$ is not derived, then $p_{C,b}$ is a disjunct of $\overline{D}_{k,b}$.

It remains to consider the case where $C \in \pi_k$ is derived. Let $D = \text{left}_\pi(C)$. The totality axiom (PHP-1) for the pigeon D states $\bigvee_{c\in[W]} p_{D,c}$. Applying the product rule (Lemma 2.1) to $p_{C,b}$ and this totality axiom, we derive

$$\bigvee_{c\in[W]} (p_{C,b} \wedge p_{D,c}), \tag{3.67}$$

which is a subformula of $\bigvee_{c\in[W]} P^{\text{left}}_{k,b,c}$. Collecting over all $C \in \pi$ using the onto axiom at b gives (3.66).

For the root block $B_{0,0}$, the translated cedent is $\bigvee_{c\in[W]} p_{C^{\text{left}}_\emptyset,c}$, which is exactly the totality axiom (PHP-1) for the pigeon $C^{\text{left}}_\emptyset$. The case for right-side pointers is identical.

The clauses of type (Ref-8) and (Ref-9). We consider the right-pointer case; the left-pointer case is analogous. For a non-root consecutive pointer, the translated cedent is

$$\overline{E}_{k,b}, \overline{P}^{\text{right}}_{k,b,c}, E_{k+1,c}. \tag{3.68}$$

From the onto axioms at holes b and c, by the product rule (Lemma 2.1), we derive

$$\bigvee_{C,D\in\pi} (p_{C,b} \wedge p_{D,c}). \tag{3.69}$$

Fix a pair $C, D \in \pi$. If $C \notin \pi_k$, then $p_{C,b}$ is a disjunct of $\overline{E}_{k,b}$. Otherwise, if $D \neq \text{right}_\pi(C)$, then $p_{C,b} \wedge p_{D,c}$ is a disjunct of $\overline{P}^{\text{right}}_{k,b,c}$. Finally, if neither case applies, then $D = \text{right}_\pi(C) \in \pi_{k+1}$ and hence $p_{D,c}$ is a disjunct of $E_{k+1,c}$. Therefore every term of the product DNF weakens to a term of (3.68), proving the desired translated axiom.

If the pointer is not between consecutive layers, then $\tau_{\pi,s}(\neg\text{rpoint}^{B_{k,b}}_{B'}) = \top$, and the translated axiom is immediate. The root and unused-block cases are also immediate from the definitions.

The clauses of type (Ref-10) and (Ref-11). These are immediate, since $\tau_{\pi,s}$ maps these axioms to the constant $\top$.

This completes the verification of item (iii) in the definition of contextual substitutions. All derivations above have size polynomial in L, D, s, n, m. The excluded-middle and incompatibility derivations also have polynomial size, and the incompatibility derivations are in fact constant-size apart from weakenings.

Finally, every positive or negative image under $\tau_{\pi,s}$ has at most $O(L^2)$ terms. Hence $\tau_{\pi,s}$ is a $(k,2)$-substitution for $\text{Ref}_s(\varphi)$ in the context of the single instance ontoPHP^L_W, in size $\text{poly}(L, D, s, n, m)$. □

It follows that Theorem 3.2 transfers to $\text{Res}(k)$ by replacing PHP by ontoPHP.

Theorem 3.6. *Let φ be a CNF formula over n variables and m clauses that admits a layered Resolution refutation π of length L and depth D. For every $k, s \in \mathbb{N}$ such that $D \leq s < L$, if $\text{Ref}_s(\varphi)$ admits a $\text{Res}(k)$-refutation of size T, then ontoPHP^L_W admits a $\text{Res}(2k)$-refutation of size at most $T \cdot \text{poly}(L, D, s, n, m) \cdot L^{O(k)}$, where $W := \lfloor s/D \rfloor$.*

Proof. By Lemma 3.5, the substitution $\tau_{\pi,s}$ is a $(k,2)$-substitution for $\text{Ref}_s(\varphi)$ in the context of ontoPHP^L_W, in size $t = \text{poly}(L, D, s, n, m)$. Moreover, every 2-DNF in the image of $\tau_{\pi,s}$ has at most $O(L^2)$ terms. Now suppose that $\text{Ref}_s(\varphi)$ admits a $\text{Res}(k)$-refutation of size T. Applying Lemma 3.4 with $\psi := \text{Ref}_s(\varphi)$, $\varphi := \text{ontoPHP}^L_W$ and $\tau := \tau_{\pi,s}$, we obtain a $\text{Res}(2k)$-refutation of ontoPHP^L_W of size at most $T \cdot t \cdot M^{O(k)} = T \cdot \text{poly}(L, D, s, n, m) \cdot L^{O(k)}$. □

4 Hardness of proving tight Resolution lower bounds

In this section we leverage the reductions within PK_d and $\text{Res}(k)$ to obtain results on the hardness of proving Resolution lower bounds. All of our results are conditioned on the hypothetical hardness of the weak pigeonhole principle in these systems.

Recall that for constant-depth Frege, we have made the following hypothesis (Hypothesis 1.4): for every $d \in \mathbb{N}$ there exist $n_0 \in \mathbb{N}$ and a function $f_d = \omega(1)$ such that for every $n \geq n_0$ the unsatisfiable CNF formula $\text{PHP}^{n^2}_n$ requires size at least $n^{f_d(n)}$ to be refuted in depth-d Frege.

The hypothesis is stated only for pigeonhole principle instances of quadratic stretch. In our applications, we often deal with instances of the form $\text{PHP}^{n^{1+\delta}}_n$ for $\delta > 1$, with possibly larger polynomial stretch. Conveniently, hardness for quadratic stretch can be amplified to any arbitrary polynomial at the cost of a constant-factor blow-up in depth. This is folklore (see, e.g., [PWW88; MPW02; Kra01]), but we include a proof in Appendix C for completeness.

Proposition 4.1 (Stretch amplification)**.** *For every pair of constants $\delta, \eta > 0$, there exists a constant $c_{\delta,\eta} \in \mathbb{N}$ such that, for every $d \in \mathbb{N}$, if $\text{PHP}^{\lceil n^{1+\delta} \rceil}_n$ admits a PK_d-refutation of size T, then $\text{PHP}^{\lceil n^{1+\eta} \rceil}_n$ admits a $\text{PK}_{d+c_{\delta,\eta}}$-refutation of size $T \cdot n^{O_{\delta,\eta}(1)}$.*

For the $\text{Res}(k)$ subsystems for constant k, to the best of our knowledge, no non-trivial upper bound is known for the weak pigeonhole principle. Here we make a significant strengthening of the hypothesis: we require exponential hardness for the onto variant with exponential stretch. More precisely, recall that Hypothesis 1.9 states that for every $k \in \mathbb{N}$ there exist constants $\alpha = \alpha(k)$ and $\beta = \beta(k)$, such that $0 < \alpha < \beta$, and $n_0 \in \mathbb{N}$ such that for every $n \geq n_0$ the unsatisfiable CNF formula $\text{ontoPHP}^{2^{\lfloor n^\alpha \rfloor}}_n$ requires size at least 2^{n^β} to be refuted in $\text{Res}(k)$.

4.1 Hardness results in constant-depth Frege

Let us start by defining a couple of classes of Resolution lower bounds for which our results apply. The first class corresponds to the natural class of formulas for which a tight exponential lower bound holds in the number of variables.

Definition 4.2 (Maximally hard formulas)**.** Let $\varphi = \{\varphi_n\}_{n\in\mathbb{N}}$ be a sequence of unsatisfiable CNF formulas where φ_n has n variables and $\mathrm{poly}(n)$ clauses. We say that φ is a *maximally hard family for Resolution* if, for every large enough n, every Resolution refutation of φ_n requires length $2^{\Omega(n)}$.

While many of the classical Resolution lower bounds the reader might have in mind fall in this category, some formulas admit non-trivial upper bounds with matching lower bounds. For example, Haken's $2^{\Omega(n)}$ lower bound for the PHP^{n+1}_n formulas [Hak85] is complemented by a $2^{O(n)}$-length upper bound by Buss and Pitassi [BP98] in depth $n^{O(1)}$; this is slightly non-trivial, since PHP^{n+1}_n has $O(n^2)$ variables, so this is not maximally hard in the sense above. The following definition aims to capture this second more general scenario.

Definition 4.3 (Naturally tight lower bounds)**.** Let $\varphi = \{\varphi_n\}_{n\in\mathbb{N}}$ be a family of unsatisfiable CNF formulas where φ_n has $\mathrm{poly}(n)$ variables and $\mathrm{poly}(n)$ clauses. We say that φ admits a *naturally tight Resolution lower bound* if there exists a function $s : \mathbb{N} \to \mathbb{N}$ such that $s(n) \geq n^{\Omega(1)}$ and constants $c \in \mathbb{N}$ and $\eta > 0$ such that for every large enough n,

(i) every Resolution refutation of φ_n requires length at least $s = s(n)$; yet,

(ii) there exists a Resolution refutation of φ_n of length at most s^c and depth $s^{1-\eta}$.

The adverb "naturally" in our definition of "naturally tight" is intended to highlight the mild bound on depth imposed by item (ii) above. This is a technical requirement for applications of our reduction (Theorem 3.2): the depth bound ensures that the number of holes in the resulting pigeonhole principle, which is proportional to s/D, remains polynomially large in s. Although this condition is not intrinsic to the notion of a tight lower bound, it is satisfied by most standard examples.

Indeed, observe that any sequence $\varphi = \{\varphi_n\}_{n\in\mathbb{N}}$ of maximally hard formulas for Resolution admits a naturally tight Resolution lower bound: the canonical tree-like Resolution refutation has length 2^{n+1} and depth n, so a $2^{\Omega(n)}$ lower bound is naturally tight. Similarly, it is worth noting that most well-known Resolution lower bounds are naturally tight in our sense [Hak85; Urq87]. On the other hand, the depth requirement in item (ii) is not automatic from the existence of a refutation of length polynomial in s. Recent (truly) supercritical size–depth trade-offs [dRFJ+25; GMRS25] demonstrate that near-optimal refutation size can be incompatible with small depth.

Our next result shows that maximally hard lower bounds are hard to derive in PK_d assuming the hardness of the weak pigeonhole principle. We state it here in a form convenient for later applications in Section 5.

Theorem 4.4. *Assume the WPHP Hypothesis (Hypothesis 1.4). For every $d \in \mathbb{N}$, every $0 < \varepsilon \leq 1$ and every $k \in \mathbb{N}$, there exists $n_0 \in \mathbb{N}$ such that for every unsatisfiable CNF formula φ over $n \geq n_0$ variables and $m \leq 3^n$ clauses, every PK_d-refutation of $\mathrm{Ref}_s(\varphi)$ for $s = \lfloor 2^{\varepsilon n} \rfloor$ has size strictly larger than s^k.*

Proof. Set $s := \lfloor 2^{\varepsilon n} \rfloor$, $W := \lfloor s/n \rfloor$ and $q := \lceil 2/\varepsilon \rceil$. For all sufficiently large n, we have $W \geq 2^{\varepsilon n}/4n$. Since $\varepsilon q \geq 2$, it follows that $W^q \geq 2^{2n}/(4n)^q \geq \max\{2^{n+1}, 3^n\}$ for all sufficiently large n. In particular, both 2^{n+1} and the number m of clauses of φ are at most W^q.

Fix the canonical tree-like Resolution refutation of φ of length 2^{n+1} and depth n. Pad it with unused clauses until its length is exactly $L := W^q$, without increasing its depth. Now suppose that

$\text{Ref}_s(\varphi)$ admits a PK_d-refutation of size T. By Theorem 3.2, there is a PK_{d+2}-refutation of $\text{PHP}_W^{W^q}$ of size at most $T \cdot \text{poly}(L, D, s, n, m) \leq T \cdot W^{O(q)}$. Here $L = W^q$ and $m \leq W^q$, and s and n are also bounded by fixed powers of W, so the polynomial overhead remains a fixed polynomial in W.

Apply Proposition 4.1 with $\delta = q - 1$ to obtain a $\mathsf{PK}_{d+2+c_\delta}$-refutation of $\text{PHP}_W^{W^2}$ of size at most $T \cdot W^{O(q)}$. Fix k_0 to be large enough such that for the value of k given by the statement we can attain $k_0 - O(q) \geq 2k$, where the constant in the term $O(q)$ is the same as in the bound $W^{O(q)}$ in the preceding line. By the WPHP Hypothesis (Hypothesis 1.4) at this fixed depth, for large enough W and hence large enough s, the formula $\text{PHP}_W^{W^2}$ requires refutations of size strictly larger than W^{k_0}. Consequently, $T > W^{k_0 - O(q)} \geq W^{2k}$. Finally, $n = O(\log s)$, so $W = \lfloor s/n \rfloor \geq s^{1/2}$ for all sufficiently large n and hence $W^{2k} \geq s^k$, getting that $T > W^{2k} \geq s^k$. This completes the proof. □

The same argument yields a hardness result for naturally tight lower bounds: instead of compressing the exponential-size tree-like refutation, it suffices to compress the best-known refutation of the formula into its matching lower bound.

Theorem 4.5. *Assume the WPHP Hypothesis holds (Hypothesis 1.4). For every $d \in \mathbb{N}$ and every sequence $\varphi = \{\varphi_n\}_{n \in \mathbb{N}}$ admitting a naturally tight Resolution lower bound of size $s = s(n)$, it holds that PK_d lacks polynomial-size refutations of the statements $\{\text{Ref}_s(\varphi_n)\}_{n \in \mathbb{N}}$ stating such a lower bound. In particular, PK_d lacks polynomial-size refutations of $2^{\Omega(n)}$ lower bounds for maximally hard formulas for Resolution.*

Proof. The proof is analogous to Theorem 4.4. The only difference is that, instead of starting from the canonical exponential-size refutation of φ_n, we may now exploit that it admits a Resolution refutation π of length $s^{O(1)}$ and depth at most $D := s^{1-\eta}$ for some η. By Proposition B.1, such a refutation can be assumed to be layered and have length $L := O(s^{O(1)} D)$ and depth D. We again invoke Theorem 3.2 to obtain refutations of PHP_W^L in PK_{d+2}, where $W = \Theta(s/s^{1-\eta}) = \Theta(s^\eta)$ and $L = s^{O(1)}$. Hypothesis 1.4 now implies a super-polynomial lower bound on the size of these refutations. □

The following is an immediate corollary, based on the fact that the pigeonhole formulas, as discussed earlier, admit a naturally tight lower bound [Hak85; BP98].

Corollary 4.6. *Assume the WPHP Hypothesis holds (Hypothesis 1.4). For every $d \in \mathbb{N}$, the system PK_d lacks polynomial-size refutations of the unsatisfiable formulas $\{\text{Ref}_{\lfloor 2^{n/20} \rfloor}(\text{PHP}_n^{n+1})\}_{n \in \mathbb{N}}$ stating the Resolution size lower bound for the pigeonhole principle.*

4.2 Hardness in Res(k): the Iterated Lower Bounds Hypothesis

Since no non-trivial lower bounds are known for $\text{ontoPHP}_n^{n^2}$ in Res(k) systems, we assumed exponential lower bounds in Hypothesis 1.9. In this regime, we obtain a hardness result much closer to that of Atserias and Müller [AM20]; recall that they showed that every Resolution refutation of $\text{Ref}_s(\varphi)$ requires size $2^{\Omega(s/n)}$ for $s \geq n^2$.

Theorem 4.7. *Assume Hypothesis 1.9. For every $k \in \mathbb{N}$ there exist constants $C = C(k)$ and $\gamma = \gamma(k) > 0$ such that for every unsatisfiable CNF formula φ over a sufficiently large number n of variables and every $s \geq n^C$, it holds that every Res(k)-refutation of $\text{Ref}_s(\varphi)$ has size at least $2^{(s/n)^\gamma}$.*

Proof. The proof follows the same template as Theorem 4.4, except building on Theorem 3.6 and Hypothesis 1.9. Let $\alpha := \alpha(2k)$ and $\beta := \beta(2k)$ be the constants given by Hypothesis 1.9, and fix $C := \lceil 2/\alpha \rceil + 2$. Let $s \geq n^C$ and set $W := \lfloor s/n \rfloor$. For all sufficiently large n, we have $W \geq n^{C-1}/2$, and hence $W^\alpha \geq n^{\alpha(C-1)}/2^\alpha \geq n + 2$; in particular, we have $2^{\lfloor W^\alpha \rfloor} \geq 2^{n+1}$. Fix π to be the canonical

tree-like Resolution refutation of φ of length at most 2^{n+1} and depth at most n, padded with dummy unused clauses until its length is exactly $L := 2^{\lfloor W^\alpha \rfloor}$.

Now suppose that $\mathsf{Res}(k)$ refutes $\textsc{Ref}_s(\varphi)$ in size T. Applying Theorem 3.6 with this choice of π, L, and W, we obtain a $\mathsf{Res}(2k)$-refutation of $\textsc{ontophp}^L_W = \textsc{ontophp}^{2^{\lfloor W^\alpha \rfloor}}_W$ in size at most $T \cdot 2^{O_k(W^\alpha)}$. Indeed, the substitution has at most $L^{O(k)}$ overhead, while all remaining polynomial factors in L, D, s, n and the number of clauses of φ are also bounded by $2^{O_k(W^\alpha)}$.

We may now invoke Hypothesis 1.9 for $\mathsf{Res}(2k)$. By choosing n large enough so that W is larger than the corresponding value m_0, we obtain $T \cdot 2^{O_k(W^\alpha)} \geq 2^{W^\beta}$. Since $\beta > \alpha$, for all sufficiently large W the $O_k(W^\alpha)$ term is at most $W^\beta/2$. Hence $T \geq 2^{W^\beta/2}$. Finally, since $W \geq s/(2n)$, for all sufficiently large n we have $T \geq 2^{\frac{1}{2}(s/2n)^\beta} \geq 2^{(s/n)^{\beta/2}}$, and so every $\mathsf{Res}(k)$-refutation of $\textsc{Ref}_s(\varphi)$ has size at least $2^{(s/n)^{\Omega(1)}}$, as desired. □

Building on the lower bound in Theorem 4.7 we now show that under the same conjectured hardness for ontoWPHP, the Iterated Lower Bound Hypothesis of Santhanam and Tzameret [ST21] holds for $\mathsf{Res}(k)$.

First, for each constant k, let us define the formulas $\textsc{Ref}^k_s(\varphi)$ generalizing $\textsc{Ref}_s(\varphi)$ for $\mathsf{Res}(k)$. The formula consists of $s+1$ blocks of variables, each encoding a line of a purported $\mathsf{Res}(k)$ refutation. Since each line in such a refutation is a disjunction over k-terms and there are at most $n^{O(k)}$ terms over n variables, for each such term t we consider variables term^B_t intended to represent that term t appears as a disjunct in line B. For example, if line B consists of the 2-DNF formula $(x_1 \wedge \neg x_2) \vee x_3 \vee (x_1 \wedge \neg x_4)$, this can be encoded by the conjunction $\mathsf{term}^B_{x_1 \wedge \neg x_2} \wedge \mathsf{term}^B_{x_3} \wedge \mathsf{term}^B_{x_1 \wedge \neg x_4} \wedge \bigwedge_{t \notin \{x_1 \wedge \neg x_2, x_3, x_1 \wedge \neg x_4\}} \neg \mathsf{term}^B_t$, where t ranges over all possible 2-terms over the variables of the given formula φ.

These are the natural generalization of lit^B_ℓ variables in $\textsc{Ref}_s(\varphi)$. Similarly, the resolution variables res^B_x are instead replaced by variables cut^B_t for each k-term t, and we update the axioms (Ref-1)-(Ref-11) accordingly. Since $\mathsf{Res}(k)$ also allows the $\wedge$-Introduction rule, we need to add some more variables to model this rule faithfully, as well as variables to indicate whether an inference is obtained by either Cut or $\wedge$-Introduction. The details are straightforward and left to the reader.

The advantage of this formalization is that a $\mathsf{Res}(k)$ lower bound implies a Resolution lower bound without any blow-up in size.

Proposition 4.8. *For every $k, r, d, s \in \mathbb{N}$, if PK^r_d admits a refutation of $\textsc{Ref}^k_s(\varphi)$ in length T, then PK^r_d admits a refutation of $\textsc{Ref}_s(\varphi)$ in length T.*

Proof. We simply exploit the fact that PK^r_d is closed under restrictions without any overhead in size. Given the refutation of $\textsc{Ref}^k_s(\varphi)$, for every $B \in [s+1]$ we restrict both the formula and the refutation by making all inferences be cuts, mapping $\mathsf{term}^B_t \mapsto \bot$ and $\mathsf{cut}^B_t \mapsto \bot$ for any term t containing two or more literals, and leaving the rest of the variables unassigned. In this way, the only term variables left correspond precisely to the lit variables of $\textsc{Ref}_s(\varphi)$, and the only cut variables are precisely the occurrences of res. □

The Iterated Lower Bounds Hypothesis of Santhanam and Tzameret [ST21, §4.1] postulates that for every reasonable[2] proof system Q, if Q is not polynomially bounded, then one may obtain a

[2]For a system Q to be "reasonable" in the sense of Santhanam and Tzameret, Q should admit polynomial-size proofs of its reflection principle. Technically speaking, neither Resolution nor k-DNF Resolution might be reasonable in this sense, since they are not known to admit short proofs of their own reflection principles [AB04; Gar19]—at least under the natural encoding [AB04]. This does not prevent the hypothesis from being meaningful for these systems—in fact, Santhanam and Tzameret prove it for Resolution [ST21, Theorem 4.7] as initial evidence for it. The main motivation for

sequence of hard formulas by composing the corresponding Ref^Q formulas with themselves. We state it formally below only[3] for the particular case of Res(k), which is the case we are interested in, using our formalization $\text{Ref}^k_s(\varphi)$.

Hypothesis 4.9 (Iterated Lower Bound Hypothesis [ST21] for Res(k))**.** *Let $k \in \mathbb{N}$, let ψ_0 be some formula over n_0 variables and $s : \mathbb{N} \to \mathbb{N}$ some nondecreasing function. For every $i \geq 1$, define*

$$\psi_i := \text{Ref}^k_{s(n_{i-1})}(\psi_{i-1}),$$

where n_i denotes the number of variables of ψ_i. The Iterated Lower Bounds Hypothesis for Res(k) *states that for every $k \in \mathbb{N}$, there exists an unsatisfiable formula ψ_0 and a nondecreasing function $s : \mathbb{N} \to \mathbb{N}$ with superpolynomial growth such that the formulas in the sequence $\{\psi_i\}_{i\in\mathbb{N}}$ are all unsatisfiable and for every $i \in \mathbb{N}$, the formula ψ_i lacks* Res(k) *refutations of length $s(n_i)$.*

Under our hypothesis on the hardness of the weak pigeonhole principle, it follows that the Iterated Lower Bound Hypothesis holds for Res(k) systems.

Theorem 4.10. *Assume Hypothesis 1.9. For $k \in \mathbb{N}$, the Iterated Lower Bound Hypothesis holds for* Res(k).

Proof. Let $C = C(k)$ and $\gamma = \gamma(k) > 0$ be the constants from Theorem 4.7. We choose $s(n) := \lceil n^{C+\log\log n} \rceil$ for all sufficiently large n, extending it arbitrarily to a nondecreasing function on $\mathbb{N}$. Clearly, s has superpolynomial growth and $s(n) \geq n^C$ for all sufficiently large n.

We first choose the initial formula. Let $\alpha = \alpha(k)$ and $\beta = \beta(k)$ be the constants from Hypothesis 1.9, and take $\varphi_0 := \text{ontoPHP}^{2^{\lfloor m^\alpha \rfloor}}_m$ for a sufficiently large m. If n_0 denotes the number of variables of φ_0, then $\log n_0 = O(m^\alpha)$, and hence $\log s(n_0) = O(m^\alpha \log m)$. On the other hand, Hypothesis 1.9 gives a size lower bound 2^{m^β} for φ_0. Since $\beta > \alpha$, and every Res(k) line over n_0 variables has size $n_0^{O(k)}$, by taking m sufficiently large we obtain that φ_0 has no Res(k)-refutation of length $s(n_0)$.

Define inductively $\varphi_{i+1} := \text{Ref}^k_{s(n_i)}(\varphi_i)$, where n_i is the number of variables of φ_i. We prove that, for every $i \geq 0$, the formula φ_i is unsatisfiable and has no Res(k)-refutation of length $s(n_i)$. The case $i = 0$ follows from our choice of φ_0.

Suppose the claim holds for i, and set $t := s(n_i)$. Since φ_i has no Res(k)-refutation with at most t lines, $\varphi_{i+1} = \text{Ref}^k_t(\varphi_i)$ is unsatisfiable. Moreover, there is a constant A, depending only on k and on the encoding, such that $n_{i+1} \leq t^A$, and φ_{i+1} has polynomially many clauses in n_{i+1}. Let π be any Res(k)-refutation of φ_{i+1}, and let ℓ be its number of lines. By Proposition 4.8, restricting π gives a Res(k)-refutation of $\text{Ref}_t(\varphi_i)$ without increasing its number of lines. Since $t = s(n_i) \geq n_i^C$, we may apply Theorem 4.7. Thus the restricted refutation has size at least $2^{(t/n_i)^\gamma}$. Since a Res(k) line over the variables of $\text{Ref}_t(\varphi_i)$ has polynomial size, and the latter formula has at most $t^{O(1)}$ variables, there is a constant B such that $\ell \cdot t^B \geq 2^{(t/n_i)^\gamma}$. For all sufficiently large n_i, this implies $\ell \geq 2^{(t/n_i)^\gamma/2}$.

It remains to compare this lower bound with $s(n_{i+1})$. By monotonicity of s and the bound $n_{i+1} \leq t^A$, we have $s(n_{i+1}) \leq s(t^A)$. For our choice $t = s(n_i) = n_i^{C+\log\log n_i + o(1)}$, we have $\log s(t^A) =$

requiring provability of reflection principles is ensuring that Pudlák's upper bound construction (Theorem 2.3) can be generalized, as stated in their Lemma 4.4. Despite the hardness of reflection principles, Lemma 4.4 does hold for Res(k) systems. This follows from [AB04, Theorem 6]: after restricting by a satisfying assignment to φ, the Res($k+1$) refutation becomes a Res(k) refutation of $\text{Ref}^k(\varphi)$.

[3] The general statement for all propositional proof systems Q would require a bit more effort to state and is not relevant here. Furthermore, we note that the original statement of Santhanam and Tzameret uses formulas $\text{Prf}^Q_s(\varphi)$ where s denotes the *size* of a purported Q-proof of φ, while in our $\text{Ref}^k_s(\varphi)$ formulas, s denotes the *length* (i.e., number of lines) of refutations. This difference is inessential, since for Res(k) systems for constant k these two measures are polynomially related, but for the sake of simplicity and precision, we opted for stating only the case for Res(k).

$O(\log n_i(\log\log n_i)^2)$, whereas $(t/n_i)^\gamma \geq n_i^{\gamma(C-1+\log\log n_i)}$. Consequently, for all sufficiently large n_i,

$$s(n_{i+1}) \leq s(t^A) < 2^{(t/n_i)^\gamma/2} \leq \ell.$$

Hence φ_{i+1} has no $\mathsf{Res}(k)$-refutation of length $s(n_{i+1})$, completing the induction.

Therefore all formulas in the sequence are unsatisfiable and require more than $s(n_i)$ lines in $\mathsf{Res}(k)$. Since s is nondecreasing and superpolynomial, this proves Hypothesis 4.9. □

5 Hardness of automating constant-depth Frege

We are ready to prove Theorem 1.6. We refer the reader to Section 1.1.4 for a discussion on the hardness of CNF-SAT.

Our proof here mimics the Atserias–Müller [AM20] strategy, with a twist. Naively applying Theorem 4.4 would mean that for every $\varepsilon > 0$ we construct the formulas $\textsc{Ref}_{2^{\varepsilon n}}(\varphi)$ and use a hypothetical automating algorithm on them. In this way, we would obtain, for every $\varepsilon > 0$, an algorithm solving CNF-SAT in time $2^{\varepsilon n} \cdot (n+m)^{O(1)}$. Instead of stopping there, we dovetail all such procedures for every $\varepsilon > 0$ together with self-reduction to obtain a single algorithm that solves CNF-SAT in time $2^{o(n)} \cdot (n+m)^{O(1)}$.

Theorem 5.1. *Assume the WPHP Hypothesis (Hypothesis 1.4). For every $d \in \mathbb{N}$, the proof system PK_d is not automatable in polynomial time on CNF formulas unless* CNF-SAT *admits a deterministic algorithm running in time $2^{o(n)} \cdot (n+m)^{O(1)}$ where n is the number of variables and m is the number of clauses of the input CNF formula.*

Proof. Suppose that A is a polynomial-time automating algorithm for PK_d. We may assume that A only outputs correct refutations, by composing it with the polynomial-time verifier for PK_d. We describe a procedure using A to solve CNF-SAT in the desired time. On input φ, normalize the formula by removing repeated literals, tautological clauses, and duplicate clauses. This preserves satisfiability and takes time polynomial in the original input length $(n+m)^{O(1)}$. Retain the original set of n variables, including any that no longer occur. The resulting formula has $m \leq 3^n$ clauses.

Observe that there is a constant $C \geq 1$ such that, for every $s \geq n$, the formula $\textsc{Ref}_s(\varphi)$ can be constructed in time at most $T(s,n,m) := (s+n+m)^C$. Furthermore, we can choose C large enough so that, by appealing to Pudlák's construction (Theorem 2.3), whenever φ is satisfiable, $A(\textsc{Ref}_s(\varphi))$ outputs a correct refutation within $T(s,n,m)$ steps.

Now, for every integer $j \geq 1$, define the following procedure B_j. On input a normalized CNF formula φ, set $s := \max\{n, \lfloor 2^{n/j} \rfloor\}$, construct the formula $\textsc{Ref}_s(\varphi)$ and run A on this formula for at most $T(s,n,m)$ steps. If A outputs a correct refutation, answer Sat; otherwise, answer Unsat.

We claim that B_j correctly decides CNF-SAT for inputs over some large enough number n_j of variables. Indeed, if φ is satisfiable, the choice of C guarantees that B_j has enough time to find a refutation that, by Pudlák's construction, must exist, and hence B_j correctly outputs Sat. On the other hand, if φ is unsatisfiable and the number of variables is large enough, then B_j correctly outputs Unsat. Indeed, for sufficiently large n, by taking $\varepsilon := 1/j$ and $s = \lfloor 2^{\varepsilon n} \rfloor$, we have that if φ is unsatisfiable, then Theorem 4.4 implies that for every constant k and every large enough n, every PK_d-refutation of $\textsc{Ref}_s(\varphi)$ has size strictly larger than s^k. Since $m \leq 3^n = s^{O(j)}$ and $n = O(j \log s)$, we have that $T(s,n,m) = s^{O(j)}$. It thus suffices to take k to be a constant such that $s^k > s^{O(j)}$ for large enough n; it then follows that B_j cannot find a refutation in time T, because none exists in size $s^{O(j)}$. It then follows that for all j, there is a threshold n_j such that B_j correctly decides every input φ with $n \geq n_j$ variables.

On inputs with $n < n_j$ variables, the procedure B_j can be wrong in exactly the following situation: it answered Sat, but in fact φ is unsatisfiable. To fix this, define a procedure $\tilde{B}_j$ as follows. On input φ, run $B_j(\varphi)$. If it answers Unsat, return Unsat. If it answers Sat, attempt to construct a satisfying assignment by self-reduction. Namely, iteratively choose literals $\ell_1, \dots, \ell_{i-1}$ and run B_j on input $\varphi \wedge \ell_1 \wedge \cdots \wedge \ell_{i-1} \wedge \neg x_i$; if the answer is Sat, set $\ell_i := \neg x_i$; otherwise, choose $\ell_i := x_i$. After assigning all variables, check the resulting assignment against φ and output Sat if it is a correct satisfying assignment, and return Unknown otherwise.

The corrected procedure $\tilde{B}_j$ does not make mistakes, although it may answer Unknown on small inputs. Indeed, its Unsat answers always are sound by construction, and its Sat answers must provide a satisfying assignment. Moreover, since the self-reduction procedure always calls B_j on the same number n of variables, whenever $n \geq n_j$, every query to B_j is answered correctly, and $\tilde{B}_j$ returns the correct answer.

The code of $\tilde{B}_j$ can be generated uniformly from j and the automating algorithm A. Each run of $\tilde{B}_j$ makes at most $n+1$ calls to B_j, so the overall runtime of $\tilde{B}_j$ is at most $(n+1) \cdot T(s, n, m+n) = (n+1) \cdot (s + 2n + m)^C$. It follows that there is an absolute constant K, independent of j, such that the running time is bounded by $c_j \cdot 2^{Kn/j}(n+m)^K$ for some constant c_j depending only on j.

To complete the argument, let B_0 denote the procedure consisting of exhaustive search on φ, and dovetail all the procedures $\tilde{B}_j$ for all $j \in \mathbb{N}$ and return the first answer of the form Sat or Unsat. Because every reported answer is sound, this defines a single sound algorithm B deciding CNF-SAT. The dovetailing incurs only a fixed polynomial simulation overhead, so there is a constant K', independent of j, such that, for every fixed j and all $n \geq n_j$, the running time of the complete algorithm B is at most $c'_j \cdot 2^{K'n/j}(n+m)^{K'}$.

It is left to argue that the running time of B is $2^{o(n)} \cdot (n+m)^{O(1)}$. It is enough to show that for every $\delta > 0$, the algorithm runs in time at most $2^{\delta n}(n+m)^{O(1)}$. Indeed, given any $\delta > 0$, choose a fixed j with $K'/j < \delta/2$. For all sufficiently large n, the constant c'_j is absorbed by $2^{\delta n/2}$, giving a running-time bound $2^{\delta n}(n+m)^{K'}$. This completes the proof. □

6 The weak pigeonhole principle is sufficient

In this section we show that concrete exponential Resolution lower bounds can be formalized at very low depth within PK when allowing substitution instances of the weak pigeonhole principle. The formalization is carried out in the U^k_d-Ind theories of Beckmann, Pudlák, and Thapen [BPT14], and then translated to constant-depth PK à la Paris–Wilkie. A definition of these theories and the propositional translation can be found in Section 2.2.

We will be careful about two issues. First, the first-order encoding of a purported Resolution refutation is chosen so that its Paris–Wilkie translation is precisely the formula $\mathrm{Ref}_s(\varphi)$ defined in Section 2.1.3. Second, we isolate the use of weak pigeonhole principle from the U^1_2-Ind reasoning, so that the increase in propositional depth caused by discharging WPHP is explicit.

Recall that the language $\mathcal{L}$ of Beckmann, Pudlák, and Thapen contains a symbol for every polynomially bounded numerical function and every numerical relation, and that BASE consists of the true universal $\mathcal{L}$-sentences. We expand $\mathcal{L}$ to $\mathcal{L}^+ := \mathcal{L} \cup \{\Pi\}$ by introducing one additional relation symbol, denoted by Π, intended to encode a purported Resolution refutation of the candidate hard formula. Recall that, under the propositional translation, U^k_{d+2}-Ind leads to refutations in PK^k_d. In particular, U^1_2-Ind yields refutations in Resolution (PK^1_0).

6.1 A first-order encoding of Ref

Let φ be a CNF formula over variables $x_1, \dots, x_n$, and let $s \in \mathbb{N}$. The relation Π in $\mathcal{L}^+$ is a ternary relation $\Pi(t, B, j)$. Its first coordinate indicates the type of the variable of $\mathrm{Ref}_s(\varphi)$, where 0 stands for enable, 1 stands for derived, etc.; its second coordinate is a block number in $[s+1]$; its third coordinate contains the remaining index of a variable of $\mathrm{Ref}_s(\varphi)$. For ease of notation, we write $\Pi(\mathsf{enable}^B)$ for $\Pi(0, B, 0)$, or $\Pi(\mathsf{lit}^B_\ell)$ for $\Pi(3, B, \ulcorner \ell \urcorner)$, where $\ulcorner \ell \urcorner \in [2n]$ codes the $2n$ possible literals over $x_1, \dots, x_n$. Similarly, $\Pi(\mathsf{lpoint}^B_{B'})$ stands for $\Pi(6, B, B')$, and so on. A similar encoding is used in [AdRKR26, §2.3].

We now define the first-order formula stating that Π encodes a Resolution refutation. The definition is obtained syntactically from the clauses (Ref-1)–(Ref-11) defining $\mathrm{Ref}_s(\varphi)$: propositional variables are replaced by the corresponding atomic instances of Π, and large disjunctions are represented using bounded existential quantifiers. For example, the first-order version of (Ref-1) is

$$\Pi(\mathsf{enable}^B) \wedge \Pi(\mathsf{res}^B_{x_j}) \wedge \Pi(\mathsf{lpoint}^B_{B'}) \wedge \Pi(\mathsf{lit}^{B'}_\ell) \;\to\; \Pi(\mathsf{lit}^B_\ell), \tag{6.1}$$

where ℓ is a literal other than x_j. The first-order version of (Ref-3) is

$$\Pi(\mathsf{enable}^B) \wedge \Pi(\mathsf{weak}^B_A) \wedge \mathsf{a\text{-}lit}^A_\ell \;\to\; \Pi(\mathsf{lit}^B_\ell), \tag{6.2}$$

where $\mathsf{a\text{-}lit}^A_\ell$ is now an $\mathcal{L}$-formula, and hence has a fixed numerical truth value once φ is fixed.

The choice axioms are represented in the same way. For example, the first-order version of (Ref-4) is

$$\Pi(\mathsf{enable}^B) \wedge \Pi(\mathsf{derived}^B) \;\to\; \exists j < n\; \Pi(\mathsf{res}^B_{x_j}), \tag{6.3}$$

while (Ref-5) becomes

$$\Pi(\mathsf{enable}^B) \wedge \neg\Pi(\mathsf{derived}^B) \;\to\; \exists A < m\; \Pi(\mathsf{weak}^B_A). \tag{6.4}$$

Similarly, the left-pointer condition is

$$\Pi(\mathsf{enable}^B) \wedge \Pi(\mathsf{derived}^B) \;\to\; \exists B' < B\; \Pi(\mathsf{lpoint}^B_{B'}). \tag{6.5}$$

We denote by $\mathrm{Ref}_\varphi(\Pi, s)$ the conjunction of bounded formulas obtained in this way from all the axioms defining $\mathrm{Ref}_s(\varphi)$. Here, we see φ as a numerical function describing a fixed sequence of CNF formulas $\{\varphi_n\}_{n\in\mathbb{N}}$; we assume that this function, given n, outputs the description of a formula φ by giving the values of the variables a-lit. Note that φ need not satisfy any conditions on uniformity, since the language $\mathcal{L}$ contains all possible functions over the natural numbers, including uncomputable ones. We later fix φ to be a concrete sequence of formulas, see Section 6.2.

Under this encoding, the Paris–Wilkie translation yields precisely the Ref formulas in Section 2.1.3. The proof is straightforward and we skip it.

Proposition 6.1. *Fix a sequence of CNF formulas $\{\varphi_n\}_{n\in\mathbb{N}}$ as above and let $n_0, s_0 \in \mathbb{N}$. The propositional formula $\langle \mathrm{Ref}_{\varphi(n)}(\Pi, s)\rangle_{n_0,s_0}$ obtained by Paris–Wilkie translation is precisely $\mathrm{Ref}_{s_0}(\varphi_{n_0})$, up to renaming of variables.*

6.2 The hard formulas and the exponential regime

We fix constants $c \geq 1$ and $\alpha > 0$ and a family $\{\varphi_n\}_{n\in\mathbb{N}}$ of unsatisfiable CNF formulas such that φ_n has at most $N_0 := cn$ variables, all the clauses have width bounded by a constant, and every

Resolution refutation requires linear width: $\mathrm{w}_{\mathsf{Res}}(\varphi_n) \geq \alpha n$. For example, one may take φ_n to be a fixed constant-degree graph pigeonhole principle over a boundary expander, for which such a linear width lower bound follows from the work of Ben-Sasson and Wigderson [BW01]. In any case, the width lower bound $\mathrm{w}_{\mathsf{Res}}(\varphi_n) \geq \alpha n$ is an external combinatorial fact for us; we do not formalize its proof inside bounded arithmetic, so the reader may feel free to choose any other φ_n satisfying the desired properties.

Next, consider the *xorification* of φ_n, denoted $\oplus\varphi_n \coloneqq \varphi_n \circ \mathrm{XOR}_2$, where every variable z_i of φ_n is replaced by $x_i \oplus y_i$ over two new variables x_i and y_i and the result is expanded into CNF form. It is easy to see that $\oplus\varphi_n$ has $N \coloneqq O(N_0) = O(n)$ variables and constant initial width.

The ambient first-order parameter in our formalization will be a number X such that $n = \lfloor \log_2 X \rfloor$. In the parlance of bounded arithmetic, "n is in Log". This is precisely the so-called *exponential-size regime* described in [AdRKR26, §1.2]. Consequently every object encoded by $O(n)$ bits has numerical magnitude $X^{O(1)}$. This includes, for instance, assignments to $\oplus\varphi_n$, clauses over its variables, and the restrictions used later in the proof.

A clause over the N variables of $\oplus\varphi_n$ is encoded by a number $c < C_0$, where $C_0 \coloneqq 2^{2N} = X^{O(1)}$. This means that a clause of the purported refutation encoded by Π can be compressed into such a number. To make this precise, let $\mathrm{Bits}(c, \ell)$ be the numerical relation asserting that the literal ℓ occurs in the encoded clause c. For a block B, define the U^1_1 formula

$$\mathrm{Code}(B, c) \;\coloneqq\; \forall \ell < 2N \, \big(\Pi(\mathrm{lit}^B_\ell) \leftrightarrow \mathrm{Bits}(c, \ell) = 1\big). \tag{6.6}$$

The following lemma states that if Π correctly encodes a Resolution refutation of an underlying formula, then the clause encoded by every block can be captured by a "code".

Lemma 6.2. *The theory* U^1_2-Ind *proves that for every block* $B \in [s+1]$ *there is a unique* $c < C_0$ *such that* $\mathrm{Code}(B, c)$ *holds.*

Proof. Uniqueness is immediate: two numbers below C_0 having the same $2N$ bits are equal, and this is a true universal $\mathcal{L}$-statement already in BASE. For existence, proceed by induction on $t \leq 2N$, maintaining a number whose first t bits agree with $\Pi(\mathrm{lit}^B_0), \ldots, \Pi(\mathrm{lit}^B_{t-1})$ and whose remaining bits are zero. More precisely, consider the formula

$$\psi(t, c, \ell) \coloneqq (\ell < t \to (\Pi(\mathrm{lit}^B_\ell) \leftrightarrow \mathrm{Bits}(c, \ell) = 1)) \wedge (t \leq \ell < 2N \to \mathrm{Bits}(c, \ell) = 0)\,, \tag{6.7}$$

and let

$$\theta(t) \coloneqq \exists c < C_0 \, \forall \ell < 2N \; \psi(t, c, \ell) \tag{6.8}$$

be the induction formula. We have that θ is an E^1_2 formula, and by the equivalence of E^1_2-Ind and U^1_2-Ind (see Section 2.2), this kind of induction is available in the theory. At stage $t+1$ we either leave the code unchanged or add the numerical value 2^t, according to the truth value of $\Pi(\mathrm{lit}^B_t)$. Since $2N = O(\log X)$ and $C_0 = X^{O(1)}$, all numerical functions used here, including 2^t, are polynomially bounded and hence belong to $\mathcal{L}$. □

6.3 A subtle trick: avoiding the width lower bound

The following observation allows us to avoid the additional quantifier alternations that would arise from formalizing a width lower bound on φ_n explicitly.

For a clause D over the variables of φ_n, let $\Lambda_n(D)$ denote the minimum possible width of a Resolution derivation of D from φ_n. To make this a total function, set $\Lambda_n(D) = N + 1$ if no such

derivation exists, and put $\Lambda_n(\top) = 0$. The function Λ is polynomially bounded, since its values are at most $N + 1 = O(n) = O(\log X)$, and it is therefore included in the language $\mathcal{L}$.

Fix $w := \lfloor \alpha n/2 \rfloor$, where recall that α is the constant such that $\mathrm{w}_{\mathsf{Res}}(\varphi_n) \geq \alpha n$. We claim that the following are true universal $\mathcal{L}$-statements and hence belong to BASE.

(W_1) If D has width less than w and is a weakening of an initial clause of φ_n, then $\Lambda_n(D) < w$.

(W_2) Suppose that D has width less than w and is a weakening either of D_1 or of a resolvent of D_1 and D_2. If $\Lambda_n(D_1) < w$ and also $\Lambda_n(D_2) < w$ in the resolved case, then $\Lambda_n(D) < w$.

(W_3) For all sufficiently large n, $\Lambda_n(\bot) \geq \alpha n > w$.

It is immediately clear that all of these statements are simply true. In particular, item (W_3) is precisely the width lower bound $\mathrm{w}_{\mathsf{Res}}(\varphi_n) \geq \alpha n$.

The important point is that no induction is used within the theory to establish these facts. The language $\mathcal{L}$ deliberately contains every polynomially bounded numerical function, and BASE contains all true universal $\mathcal{L}$-sentences, so item (W_1), item (W_2) and item (W_3) are in BASE and can be used in our proofs. With the width lower bound taken care of, we turn our attention to the restriction argument.

6.4 The restriction argument as the weak pigeonhole principle

Let us spell out the counting argument behind the "random" restriction argument. Recall that N_0 denotes the number of variables of φ_n. In the context of xorified formulas $\oplus\varphi_n$, a restriction ρ chooses, independently for every $j \in [N_0]$, one of the four possibilities

$$x_j \mapsto 0,\ y_j \mapsto *; \qquad x_j \mapsto 1,\ y_j \mapsto *; \qquad x_j \mapsto *,\ y_j \mapsto 0; \qquad x_j \mapsto *,\ y_j \mapsto 1, \tag{6.9}$$

where $*$ indicates that the variable is left unassigned. We refer to this choice as the *restriction digit* of ρ at coordinate j. A restriction is therefore encoded by a number $\rho < Q$, where $Q := 4^{N_0}$. Observe that $\oplus\varphi_n \restriction_\rho$ is just φ_n, up to renaming of variables.

For a clause code c, let $\mathrm{Supp}(c)$ denote the number of indices $j \in [N_0]$ for which the encoded clause mentions at least one variable indexed by j (one of x_j or y_j). Let $\mathrm{True}(c, \rho)$ mean that the restriction ρ satisfies the encoded clause, and let $\mathrm{Rest}(c, \rho)$ be $\top$ if $\mathrm{True}(c, \rho)$ holds, and otherwise the code of the surviving clause after the restriction. All three are numerical objects and live in $\mathcal{L}$.

Now fix a clause code c, and suppose that $\mathrm{Supp}(c) = h$. At every coordinate j belonging to the support of c, the clause contains at least one literal over the variables $\{x_j, y_j\}$. Among the four possible restriction digits at j, at least one makes such a literal true. For example, if x_j occurs positively in the clause, then the digit $x_j \mapsto 1, y_j \mapsto *$ already satisfies the clause; if $\neg y_j$ occurs, then the digit $x_j \mapsto *, y_j \mapsto 0$ does so. Consequently, if a restriction ρ does *not* satisfy the clause encoded by c, then at every supported coordinate it has at most three possible digits. At a coordinate outside the support, on the other hand, all four digits remain possible.

It follows that the number of restrictions which do *not* satisfy c is at most $3^h 4^{N_0 - h}$. More precisely, there is an injective encoding of all such restrictions into an interval of this size. Fix an ordering of the coordinates. At a supported coordinate j, order the restriction digits which do not already satisfy the clause and encode the one chosen by ρ by a symbol in $\{0, 1, 2\}$. At an unsupported coordinate, retain the original digit in $\{0, 1, 2, 3\}$. Reading the resulting string—base 3 at supported coordinates and base 4 elsewhere—gives an injective map ι of the form

$$\iota : \left\{\rho \in 4^{N_0} \mid \neg\,\mathrm{True}(c, \rho)\right\} \to \left[3^h 4^{N_0 - h}\right]. \tag{6.10}$$

In particular, if $h \geq w$, then, since $3/4 < 1$,

$$3^h 4^{N_0-h} = 4^{N_0}\left(\frac{3}{4}\right)^h \leq 4^{N_0}\left(\frac{3}{4}\right)^w = Q\left(\frac{3}{4}\right)^w. \tag{6.11}$$

We therefore set $B_0 := Q \cdot (3/4)^w = 3^w 4^{N_0-w}$ and every restriction that fails to satisfy a clause of support at least w can be encoded by a number below B_0.

We denote this encoding ι by the function $\mathrm{Rank}(c, \rho)$. Conversely, let $\mathrm{Unrank}(c, b)$ decode b according to the same convention; on values b which do not encode a valid restriction we may define $\mathrm{Unrank}(c, b)$ arbitrarily. What matters is that on inputs c and ρ such that $\neg\,\mathrm{True}(c, \rho)$ and $\mathrm{Supp}(c) \geq w$, we have that $\mathrm{Rank}(c, \rho)$ is a well-defined total function such that $\mathrm{Rank}(c, \rho) < B_0$ and $\mathrm{Unrank}(c, \mathrm{Rank}(c, \rho)) = \rho$. In particular, $\mathrm{Rank}(c, \cdot)$ is injective on the restrictions which do not satisfy c.

Both Rank and Unrank are obtained by scanning the $N_0 = O(n)$ coordinates and performing elementary arithmetic. Their inputs and outputs have $O(n)$ bits, and the functions are polynomial-time computable and polynomially bounded in the ambient parameter X. We may therefore use them as numerical functions of the language $\mathcal{L}$. The properties of Rank and Unrank stated above are true universal $\mathcal{L}$-sentences, and hence are available in BASE.

We now define the relation which will occur in the weak pigeonhole principle. Its "pigeons" are the possible restrictions $\rho < Q$ and its "holes" are pairs (B, b) with $B \in [s+1]$ and $b < B_0$. Let $M := (s+1)B_0$ denote the set of holes. We define the following relation Γ between Q and M as

$$\begin{aligned}\Gamma(\rho, (B, b)) \;:=\; \exists c < C_0 \,\big(&\Pi(\mathsf{enable}^B) \wedge \mathrm{Code}(B, c) \wedge \mathrm{Supp}(c) \geq w \\ &\wedge \neg\,\mathrm{True}(c, \rho) \wedge \mathrm{Unrank}(c, b) = \rho\big).\end{aligned} \tag{6.12}$$

Thus (B, b) certifies that block B is enabled, contains a wide clause, and that ρ is one of the restrictions not satisfying this clause. Observe that Γ is an E^1_2 formula, since Code is U^1_1. The following lemma shows that the map defined by Γ is total and injective.

Lemma 6.3 (Totality and injectivity of Γ)**.** *The theory* U^1_2-Ind *proves the following properties of* Γ*.*

(*i*) *Totality: If block B is enabled in Π, its clause has support at least w, and a restriction ρ does not satisfy it, then there is $b < B_0$ such that $\Gamma(\rho, (B, b))$ holds.*

(*ii*) *Injectivity: If both $\Gamma(\rho, (B, b))$ and $\Gamma(\rho', (B, b))$ hold, then $\rho = \rho'$.*

Proof. For the first item, use Lemma 6.2, available in U^1_2-Ind, to obtain the unique code c of the contents of block B. Since $\mathrm{Supp}(c) \geq w$ and $\neg\,\mathrm{True}(c, \rho)$, we can use the function Rank to obtain some $b = \mathrm{Rank}(c, \rho) < B_0$ such that $\mathrm{Unrank}(c, b) = \rho$, so $\Gamma(\rho, (B, b))$ holds.

For the second item, let c and c' witness the two images of Γ on ρ and ρ'. Both $\mathrm{Code}(B, c)$ and $\mathrm{Code}(B, c')$ hold, so the uniqueness of codes from Lemma 6.2 gives $c = c'$. Since $\mathrm{Unrank}(c, b)$ is a function, it can only have a unique value, and hence $\rho = \rho'$. □

To complete the argument, it now suffices to invoke the weak pigeonhole principle on the relation defined by Γ. This will imply there exists a pigeon that cannot fly into any hole, and such a pigeon is precisely a *good restriction* ρ, meaning that ρ succeeds in satisfying all clauses with large enough support.

Let us first determine the stretch required for the instance of the pigeonhole principle. Let $\zeta := \log_2(4/3)$ and fix a rational constant δ such that $0 < \delta < \alpha\zeta/2$, and set $s := \lfloor 2^{\delta n} \rfloor$. Recall that Q

is the total number of pigeons (restrictions) and M denotes the total number of holes (pairs (B, b) witnessing failures), so we have

$$\frac{M}{Q} = \frac{(s+1)B_0}{Q} = \frac{(s+1)Q(3/4)^w}{Q} = (s+1)\left(\frac{3}{4}\right)^w \tag{6.13}$$

and since $s = \lfloor 2^{\delta n} \rfloor$ and $w = \lfloor \alpha n/2 \rfloor$, we obtain, for all sufficiently large n,

$$(s+1)\left(\frac{3}{4}\right)^w \leq 2^{\delta n+1}\left(\frac{3}{4}\right)^w \leq 2^{\delta n+1-\zeta w} \leq 2^{\delta n+1-\zeta(\alpha n/2-1)} = 2^{-\kappa n+1+\zeta} \leq 2^{-\kappa n/2}, \tag{6.14}$$

where we let $\kappa := \alpha\zeta/2 - \delta > 0$, and since $1+\zeta$ is a constant, we have $1+\zeta \leq \kappa n/2$ for large enough n, which justifies the last inequality.

Since $Q = 4^{N_0}$ and $N_0 = Kn$ for some constant $K \geq 1$, we have $Q = 2^{2Kn}$. It follows that if we set $\eta := \kappa/(4K) > 0$, then $M \leq Q^{1-\eta}$ for all sufficiently large n. Fixing a rational constant $1 < c < 1/(1-\eta)$, it follows that $\lceil M^c \rceil \leq Q$ for all sufficiently large n. This means that our map defined by Γ sends $Q \geq \lceil M^c \rceil$ pigeons into M holes, precisely in the range of the weak pigeonhole principle.

Now, for a binary relation $R(p, h)$ and a rational $c > 1$, write $\mathrm{WPHP}_c(R; m)$ for the relational form of the ordinary injective weak pigeonhole principle asserting that if R is total, then it cannot be injective,

$$\Big(\forall p < \lceil m^c \rceil\, \exists h < m\; R(p,h)\Big) \;\rightarrow\; \exists p < p' < \lceil m^c \rceil\, \exists h < m\; \big(R(p,h) \wedge R(p',h)\big). \quad (\mathrm{WPHP}_c(R;m))$$

We do not impose functionality on R. The negation of this formula, after Paris–Wilkie translation, is precisely the usual CNF unary encoding of the weak pigeonhole principle with $\lceil m^c \rceil$ pigeons and m holes.

Let us now say that a formula ϕ is U^r_ℓ*-definable over* U^k_d-Ind if there exists a U^r_ℓ-formula ψ over the same variables as ϕ such that U^k_d-Ind $\vdash \forall x\, (\phi(x) \leftrightarrow \psi(x))$. In what follows, we denote by U^k_d-Ind + WPHP(U^r_ℓ) the theory obtained by expanding U^k_d-Ind with instances of the formulas $\mathrm{WPHP}_c(R)$ above for any rational $c \in \mathbb{Q}$ with $c > 1$ and relations R that are U^r_ℓ-definable over U^k_d-Ind.

Note that the formula defining the relation Γ in (6.12) is a E^1_2-formula, so before proceeding any further, we show that Γ is in fact U^1_2-definable. For a block B, a clause code c, and a literal ℓ, define

$$\mathrm{Mismatch}(B, c, \ell) := \big(\mathrm{Bits}(c,\ell) = 1 \wedge \neg\Pi(\mathrm{lit}^B_\ell)\big) \vee \big(\mathrm{Bits}(c,\ell) = 0 \wedge \Pi(\mathrm{lit}^B_\ell)\big). \tag{6.15}$$

stating that c disagrees with the contents of block B at literal ℓ. Observe that $\mathrm{Code}(B, c) \equiv \forall \ell < 2N\, \neg\,\mathrm{Mismatch}(B, c, \ell)$.

Consider now the formula

$$\begin{aligned}\widehat{\Gamma}(\rho, (B, b)) := \forall c < C_0 + 1\; \exists \ell < 2N\; \Big[&\big(c = C_0 \wedge \Pi(\mathsf{enable}^B)\big) \\ &\vee \Big(c < C_0 \wedge \big(\mathrm{Mismatch}(B, c, \ell) \vee H(c, \rho, b)\big)\Big)\Big],\end{aligned} \tag{6.16}$$

where $H(c, \rho, b) := \mathrm{Supp}(c) \geq w \wedge \neg\,\mathrm{True}(c, \rho) \wedge \mathrm{Unrank}(c, b) = \rho$.

This is a U^1_2-formula. It is left to prove that it is equivalent to Γ.

Lemma 6.4. *The theory* U^1_2-Ind *proves that* Γ *and* $\widehat{\Gamma}$ *are equivalent. That is, the relation defined by* Γ *is* U^1_2*-definable over* U^1_2-Ind.

Proof. Working within U^1_2-Ind, by Lemma 6.2, U^1_2-Ind proves that every block B has a unique code $c_B < C_0$. For this code, every mismatch is false, so $\widehat{\Gamma}(\rho, (B, b))$ implies

$$\mathrm{Supp}(c_B) \geq w \wedge \neg\,\mathrm{True}(c_B, \rho) \wedge \mathrm{Unrank}(c_B, b) = \rho. \tag{6.17}$$

Together with $\Pi(\mathsf{enable}^B)$, this is precisely $\Gamma(\rho, (B, b))$.

Conversely, suppose $\Gamma(\rho, (B, b))$ holds. By the uniqueness of codes from Lemma 6.2, the code of B must be c_B. For every other code $c < C_0$, there exists $\ell < 2N$ such that $\mathrm{Mismatch}(B, c, \ell)$ holds. For $c = c_B$, the remaining conditions hold by the definition of Γ. Hence $\widehat{\Gamma}(\rho, (B, b))$ holds. □

We are now ready to prove the main result of this section. The argument has two parts. First, we use the weak pigeonhole principle for the relation Γ to obtain a restriction under which every enabled clause is either satisfied or has small support. Second, we show by induction on the blocks of the purported refutation that such a restriction contradicts the Resolution width lower bound for φ_n. The latter argument uses only U^1_2-Ind, so the weak pigeonhole principle is invoked exactly once.

Theorem 6.5 (Formalized exponential Resolution lower bound)**.** *There exists* $\delta > 0$, $X_0 \in \mathbb{N}$ *and a family of constant-width unsatisfiable CNF formulas* $\{\psi_n\}_{n\in\mathbb{N}}$ *on* $O(n)$ *variables such that for the functions* $n(X) := \lfloor \log_2 X \rfloor$ *and* $s(X) := \lfloor 2^{\delta n(X)} \rfloor$ *in the language* $\mathcal{L}$ *of* U^1_2-Ind*, it holds that*

$$\mathsf{U}^1_2\text{-Ind} + \mathrm{WPHP}(\mathsf{U}^1_2) \vdash \forall X \geq X_0\, \neg\,\mathrm{Ref}_{\psi_{n(X)}}(\Pi, s(X)).$$

Furthermore, the proof uses only one substitution instance of the weak pigeonhole principle.

Proof. The formulas ψ_n are the formulas $\oplus\varphi_n$ used throughout this section: xorified instances of the graph pigeonhole principle over a suitable constant-degree bipartite expander (see Section 6.2). Our argument works for large enough n, which fixes X_0 in the statement; this can be computed from the rest of the argument.

Now, fix the parameters δ, w, and $c > 1$ as defined earlier in this section, so that $\lceil M^c \rceil \leq Q$ for all sufficiently large n, and suppose for contradiction $\mathrm{Ref}_{\oplus\varphi_n}(\Pi, s)$ holds, meaning that Π encodes a correct Resolution refutation in at most $s = s(n)$ clauses. We derive a contradiction in two steps.

The restriction argument. We call a restriction ρ *good* if every enabled block is either satisfied by ρ or has support less than w. To express this within the required quantifier complexity, let

$$\begin{aligned}\mathrm{Good}(\rho) \;:=\; \forall B \leq s+1\, \forall c < C_0\, \exists \ell < 2N\, \big(&\neg\Pi(\mathsf{enable}^B) \vee \mathrm{Mismatch}(B, c, \ell) \\ &\vee \mathrm{True}(c, \rho) \vee \mathrm{Supp}(c) < w\big).\end{aligned} \tag{6.18}$$

This is a U^1_2-formula. By Lemma 6.2, every block has a unique content code, so $\mathrm{Good}(\rho)$ expresses precisely the intended condition.

We claim that a good restriction exists. Suppose otherwise, meaning that $\forall \rho < Q\, \neg\,\mathrm{Good}(\rho)$ holds. This means, in particular, that for $\rho < \lceil M^c \rceil \leq Q$ we also have $\neg\,\mathrm{Good}(\rho)$. By definition of Good this yields a block B and a code $c < C_0$ such that B is enabled, its code is c, it has support at least w and $\neg\,\mathrm{True}(c, \rho)$ holds.

Now, by Lemma 6.2, there exists a hole $h < M$ such that $\Gamma(\rho, h)$ holds. Consequently,

$$\forall \rho < \lceil M^c \rceil\, \exists h < M\, \Gamma(\rho, h). \tag{6.19}$$

By Lemma 6.4, the relation Γ is U^1_2-definable over U^1_2-IND. We may therefore invoke the corresponding instance of WPHP(U^1_2), obtaining distinct restrictions $\rho, \rho' < \lceil M^c \rceil$ together with a hole $h < M$ such that both $\Gamma(\rho, h)$ and $\Gamma(\rho', h)$ hold. This contradicts the injectivity of Γ proven in Lemma 6.3. It follows that there exists a good ρ.

The width lower bound. Let ρ be any such good restriction. We show by induction on the length $s+1$ of the purported refutation that every enabled block has restricted width measure less than w.

For a block B, define

$$\mathrm{Low}_\rho(B) \coloneqq \forall c < C_0\ \exists \ell < 2N\ \big(\mathrm{Mismatch}(B, c, \ell) \vee \Lambda_n(\mathrm{Rest}(c, \rho)) < w\big). \tag{6.20}$$

This is a U^1_2-formula. If c is the actual content code of B, then every mismatch is false, so $\mathrm{Low}_\rho(B)$ asserts that the restricted clause has width less than w.

We shall prove by U^1_2-IND-induction on B the formula

$$\forall B' \le B\ \big(\neg\Pi(\mathsf{enable}^{B'}) \vee \mathrm{Low}_\rho(B')\big). \tag{6.21}$$

Note that the induction formula is again a U^1_2-formula, and recall that the empty clause is encoded by the last block in Π, indexed by $B = s+1$.

Suppose the formula holds for all blocks preceding B. If B is disabled, then we are done. Otherwise, fix a code $c < C_0$. If $\mathrm{Code}(B, c)$ fails, some literal witnesses a mismatch, and $\mathrm{Low}_\rho(B)$ follows. We may therefore assume that c is the actual content code of B. If $\mathrm{True}(c, \rho)$ holds, then $\mathrm{Rest}(c, \rho) = \top$ and as discussed in Section 6.3 we have $\Lambda_n(\top) = 0 < w$. Otherwise, since ρ is good, $\mathrm{Supp}(c) < w$. In particular, the restricted clause has ordinary width less than w.

We now use the local correctness of the refutation encoded by Π. By the axioms of Ref, every enabled block is either a weakening of an initial clause or is obtained by a Resolution inference followed by weakening. In the first case, restriction leaves either a satisfied clause or a weakening of an initial clause of φ_n, up to the canonical signed renaming. By property (W_1) of Λ_n, we have $\Lambda_n(\mathrm{Rest}(c, \rho)) < w$.

In the second case, let $B_L, B_R < B$ be the enabled predecessor blocks, and let c_L, c_R be their unique content codes, whose existence follows from Lemma 6.2. By the induction hypothesis, $\Lambda_n(\mathrm{Rest}(c_L, \rho)) < w$ and $\Lambda_n(\mathrm{Rest}(c_R, \rho)) < w$. Restrictions preserve the local Resolution inference in the following sense: the restricted conclusion is either a weakening of a surviving predecessor or a weakening of the resolvent of the two surviving predecessors. This follows by considering whether the pivot variable is fixed by ρ or remains unset, and is a true universal numerical statement available in BASE. By property (W_2) of Λ_n, we again obtain $\Lambda_n(\mathrm{Rest}(c, \rho)) < w$. This completes the induction.

Finally, consider (6.21) for the distinguished final block $B = s+1$ of Π. Since Π is supposed to encode a correct refutation, this block is enabled and contains no literals. Since restriction preserves the empty clause, we have that (6.21) gives $\Lambda_n(\bot) < w$. This contradicts property (W_3), namely $\Lambda_n(\bot) \ge w$, completing the proof. □

6.5 Propositional consequences

We finally translate the preceding first-order argument back to propositional proof systems. The important point is that all the reasoning outside the weak pigeonhole principle is formalized in U^1_2-IND and therefore translates into Resolution. The only increase in depth comes from substituting

the formulas defining the relation Γ in (6.12) and (6.16) into a proof of the weak pigeonhole principle.

Theorem 6.6 (Propositional upper bound assuming WPHP)**.** *There exist constants $\delta > 0$ and $1 < c < 2$ and a family of constant-width unsatisfiable CNF formulas $\{\psi_n\}_{n\in\mathbb{N}}$ over $O(n)$ variables such that for $s(n) := \lfloor 2^{\delta n} \rfloor$ and every $d \in \mathbb{N}$, if the formulas $\{\mathrm{PHP}_m^{\lceil m^c \rceil}\}_{m\in\mathbb{N}}$ admit PK_d-refutations of size $T(m)$, then the unsatisfiable formulas $\{\textsc{Ref}_{s(n)}(\psi_n)\}_{n\in\mathbb{N}}$ admit $\mathsf{PK}_{d+1+1/2}$-refutations of size $\mathrm{poly}(s(n)) \cdot T(\mathrm{poly}(s(n)))$. Furthermore, if the refutations of the weak pigeonhole principle are* **P***-uniform, then so are the refutations of $\textsc{Ref}_{s(n)}(\psi_n)$.*

Proof. By the propositional translation in Theorem 2.4 and Proposition 6.1 applied to the formalization in Theorem 6.5, all the derivations except for the application of WPHP translate into polynomial-size Resolution derivations. Indeed, every induction formula occurring in the formalization belongs to U^1_2. It therefore remains to discharge the translated instances of WPHP of the form $\mathrm{WPHP}_c(\widehat{\Gamma}; M)$, where $M \leq 4^{O(n)} = 2^{O(n)} = s(n)^{O(1)}$.

Fix a pigeon ρ and a hole $h = (B, b)$ and recall that

$$\widehat{\Gamma}(\rho, h) := \Pi(\mathsf{enable}^B) \wedge \forall c < C_0\ \exists \ell < 2N \left(\mathrm{Mismatch}(B, c, \ell) \vee H(c, \rho, b)\right), \tag{6.22}$$

where

$$H(c, \rho, b) := \mathrm{Supp}(c) \geq w \wedge \neg\, \mathrm{True}(c, \rho) \wedge \mathrm{Unrank}(c, b) = \rho. \tag{6.23}$$

Since H is purely numerical, its truth value is determined by the parameters. Let $C_{\rho,b} := \{c < C_0 \mid \neg H(c, \rho, b)\}$ and observe that after evaluating all numerical predicates, the Paris–Wilkie translation of $\widehat{\Gamma}$ is of the form

$$\langle\widehat{\Gamma}(\rho, h)\rangle = \mathsf{enable}^B \wedge \bigwedge_{c\in C_{\rho,b}} \bigvee_{\ell<2N} L^B_{c,\ell}, \tag{6.24}$$

where $L^B_{c,\ell} := \mathsf{lit}^B_\ell$ if $\mathrm{Bits}(c, \ell) = 0$ and $L^B_{c,\ell} := \neg\mathsf{lit}^B_\ell$ otherwise. Thus, the translated formula is in CNF. Each clause has width at most $2N$, and there are at most C_0 clauses. Since $2N = O(n) = O(\log s(n))$ and $C_0 = 2^{2N} = s(n)^{O(1)}$, the translation has depth 2 where the innermost connective is of width $O(\log s(n))$.

We now substitute $\langle\widehat{\Gamma}(\rho, h)\rangle$ for the atoms $p_{\rho,h}$ in a PK_d-refutation of $\mathrm{PHP}_M^{\lceil M^c \rceil}$. The pigeon clauses become $\bigvee_{h<M}\langle\widehat{\Gamma}(\rho, h)\rangle$, while the injectivity clauses become $\neg\langle\widehat{\Gamma}(\rho, h)\rangle \vee \neg\langle\widehat{\Gamma}(\rho', h)\rangle$. These are precisely the propositional translations of the clauses expressing $\mathrm{WPHP}_c(\widehat{\Gamma}; M)$.

Each substituted formula has size $s(n)^{O(1)}$, and the original refutation has size $T(M)$. Thus, the substituted derivation has size at most $s(n)^{O(1)} \cdot T(M)$. Combining this derivation with the polynomial-size Resolution derivation coming from the rest of the argument in Theorem 6.5 gives a PK_{d+2}-refutation of $\textsc{Ref}_{s(n)}(\psi_n)$ of size $\mathrm{poly}(s(n)) \cdot T(\mathrm{poly}(s(n)))$. Since the bottom fan-in of the formula is $O(\log s(n))$, we have that this is in fact a $\mathsf{PK}_{d+1+1/2}$ refutation.

Let us finally verify that the uniformity claim follows. First, the formulas ψ_n can be efficiently constructed by using well-known explicit constructions of bipartites expander (see, for example, [CRVW02]). As for the proofs themselves, in general, refutations obtained from translation over the U^k_d-$\textsc{Ind}$ theories may not be uniform, since proofs may appeal to any non-computable function in the base language $\mathcal{L}$. However, by inspecting the formalization in Theorem 6.5, we observe that all numerical functions used in the proof are computable in polynomial time, except for the function $\Lambda_n(D)$ computing the minimum width of a Resolution derivation of D. Nevertheless, the function Λ_n can be computed in time $2^{O(n)}$, and since the size of the formula $\textsc{Ref}_s(\psi_n)$ is $2^{\Theta(n)}$, we can afford to compute Λ_n explicitly and obtain **P**-uniform proofs. □

The previous results obtain propositional proofs of exponential-size lower bounds, but we can easily obtain fixed-polynomial lower bounds at the cost of further increasing the depth of the refutation by an additive constant. To do this, we exploit a technique introduced in [AdRKR26, Theorem 8.3].

Theorem 6.7. *There exists a sequence of unsatisfiable CNF formulas $\{\gamma_n\}_{n\in\mathbb{N}}$ over $\mathrm{poly}(n)$ variables and $\mathrm{poly}(n)$ clauses such that for every $k, d \in \mathbb{N}$, if PK_d admits polynomial-size refutations of $\{\mathrm{PHP}_m^{m^2}\}_{m\in\mathbb{N}}$, then $\mathsf{PK}_{d+O(1)}$ admits polynomial-size refutations of the unsatisfiable formulas $\{\mathrm{Ref}_{n^k}(\gamma_n)\}_{n\in\mathbb{N}}$.*

Proof. Let $\delta > 0$ and $1 < c < 2$ be the constants from Theorem 6.6. Let d_0 be a fixed upper bound on the left degrees of the bipartite graphs G_N underlying the formulas ψ_N from that theorem. We first describe how to construct a family $\{F_n\}_{n\in\mathbb{N}}$ of polynomial-size CNF formulas and set $\gamma_n := \oplus F_n$.

The variables of F_n are $p_{i,j}$, for $i \in [n+1]$ and $j \in [n]$, together with selector variables $z_{i,t,u}$, for $i \in [n+1]$, $t \in [d_0]$, and $u \in [\ell_n]$, where $\ell_n := \lceil \log_2 n \rceil$. Fix a surjection $\mathrm{dec}_n : \{0,1\}^{\ell_n} \to [n]$. For each i, write $\mathbf{z}_i = (z_{i,t,u})_{t,u}$. For a bit string $a \in \{0,1\}^{d_0\ell_n}$, write $a = (a^{(1)}, \ldots, a^{(d_0)})$ with $a^{(t)} \in \{0,1\}^{\ell_n}$ and put $h_t(a) := \mathrm{dec}_n(a^{(t)})$. Let $\mathbf{z}_i \neq a$ denote the clause

$$\bigvee_{t\in[d_0]} \bigvee_{u\in[\ell_n]} (z_{i,t,u} \neq a_u^{(t)}), \tag{6.25}$$

where $(z \neq 0)$ abbreviates the literal z and $(z \neq 1)$ abbreviates $\neg z$. Define F_n to consist of the clauses

$$(\mathbf{z}_i \neq a) \vee \bigvee_{t\in[d_0]} p_{i,h_t(a)} \qquad \text{for } i \in [n+1],\ a \in \{0,1\}^{d_0\ell_n}, \tag{6.26}$$

together with the usual injectivity clauses

$$\neg p_{i,j} \vee \neg p_{i',j} \qquad \text{for } i < i' \leq n+1,\ j \in [n]. \tag{6.27}$$

For every assignment to the selectors, the first group of clauses requires each pigeon to occupy at least one hole; the second group prevents two distinct pigeons from occupying the same hole. Hence F_n is unsatisfiable. Furthermore, F_n has $\mathrm{poly}(n)$ variables and clauses, each clause has width at most $d_0\ell_n + d_0 = O(\log n)$, and consequently its xorification $\gamma_n = \oplus F_n$, expanded into CNF, also has $\mathrm{poly}(n)$ variables and clauses.

We claim that for every $N \leq n$ there exists a restriction $\rho_{n,N}$ such that

$$\gamma_n \restriction_{\rho_{n,N}} \equiv \oplus\mathrm{GPHP}(G_N) = \psi_N, \tag{6.28}$$

where equivalence means equality after removing satisfied clauses and duplicate literals and clauses, and renaming the remaining variables. First work with F_n. Keep pigeons $[N+1]$ and holes $[N]$. For each pigeon $i \in \{N+2, \ldots, n+1\}$, set $p_{i,i-1} = 1$ and all other variables $p_{i,j}$ to 0; these pigeons are matched to the holes $\{N+1, \ldots, n\}$. For each $i \in [N+1]$, set $p_{i,j} = 0$ whenever (i,j) is not an edge of G_N, leaving the edge variables free.

For each surviving pigeon $i \in [N+1]$, fix its selector vector $\mathbf{z}_i$ to encode a list of exactly d_0 holes whose set of entries is its neighborhood in G_N; repeat entries if necessary. For a matched pigeon $i > N+1$, fix $\mathbf{z}_i$ so that at least one of its selected holes is $i-1$. Under this restriction, for each pigeon precisely one selector clause is not already satisfied by its selector literals. For a surviving pigeon, that clause becomes exactly its totality clause; for a matched pigeon, it is satisfied by the matched edge. All remaining injectivity clauses are precisely those of $\mathrm{GPHP}(G_N)$. Thus the restricted F_n is $\mathrm{GPHP}(G_N)$.

Finally, lift the restriction to the xorification by fixing both variables of each XOR gadget representing a fixed input variable to constants of the prescribed parity, and leaving both variables of every unfixed edge gadget free. XORification commutes with this restriction, up to removing satisfied clauses, yielding the claim.

Now fix $k \in \mathbb{N}$ and set $N := \lceil (k/\delta) \log_2 n \rceil + 1$. For sufficiently large n, we have $N \leq n$ and $n^k \leq s(N) := \lfloor 2^{\delta N} \rfloor = n^{O(k)}$. Fix the restriction $\rho = \rho_{n,N}$ as defined above. Since Resolution is closed under restrictions, a refutation of γ_n of length at most n^k restricts to a refutation of ψ_N of length at most n^k. Furthermore, Resolution itself proves that the system is closed under restrictions, meaning that there are polynomial-size Resolution derivations of the form

$$\textsc{Ref}_t(\varphi, \pi) \wedge (\pi' = \pi_{\restriction \rho}) \wedge (\varphi' = \varphi_{\restriction \rho}) \vdash_{\mathsf{Res}} \textsc{Ref}_t(\varphi', \pi'), \tag{6.29}$$

where, once ρ is fixed, $(\pi' = \pi_{\restriction \rho}) \wedge (\varphi' = \varphi_{\restriction \rho})$ are depth-2 formulas. See [AdRKR26, Proposition 6.1] for a proof. It then follows that

$$\textsc{Ref}_{n^k}(\gamma_n, \pi) \wedge (\pi' = \pi_{\restriction \rho}) \vdash^{n^{O(k)}}_{\mathsf{Res}} \textsc{Ref}_{s(N)}(\psi_N, \pi'). \tag{6.30}$$

Now suppose that the formulas $\{\mathrm{PHP}^{m^2}_m\}_{m \in \mathbb{N}}$ admit polynomial-size PK_d-refutations. By stretch amplification as in Proposition 4.1, the same holds for the polynomial stretch appearing in Theorem 6.6, with at most a constant increase in depth. Therefore, we have that

$$\textsc{Ref}_{s(N)}(\psi_N, \pi') \vdash^{n^{O_k(1)}}_{\mathsf{PK}_{d+O(1)}} \bot. \tag{6.31}$$

Combining this with (6.30), we obtain

$$\textsc{Ref}_{n^k}(\gamma_n, \pi) \wedge (\pi' = \pi_{\restriction \rho}) \vdash^{n^{O(k)}}_{\mathsf{PK}_{d+O(1)}} \bot. \tag{6.32}$$

It remains to eliminate the auxiliary variables encoding π'. Since ρ is fixed, these definitions have polynomial size and depth at most 2, and substituting them into the preceding refutation we obtain a $\mathsf{PK}_{d+O(1)}$-refutation of $\textsc{Ref}_{n^k}(\gamma_n)$ of size $n^{O(k)}$. This completes the proof. □

Note that all pigeonhole principles admit polynomial-size proofs in $\mathbf{TC}^0$-Frege systems. As a result, Theorem 6.6 and Theorem 6.7 immediately imply that Resolution lower bounds in both the exponential and polynomial-size regimes admit polynomial-size proofs in $\mathbf{TC}^0_d$-Frege for a large enough constant d.

Corollary 6.8. *There exist constants $d \in \mathbb{N}$ and $\delta > 0$ and sequences of unsatisfiable formulas $\{\psi_n\}_{n \in \mathbb{N}}$ and $\{\gamma_n\}_{n \in \mathbb{N}}$ over $\mathrm{poly}(n)$ variables and $\mathrm{poly}(n)$ clauses such that $\mathbf{TC}^0_d$-Frege refutes in polynomial size the unsatisfiable formulas $\{\textsc{Ref}_{\lfloor 2^{\delta n} \rfloor}(\psi_n)\}_{n \in \mathbb{N}}$ and $\{\textsc{Ref}_{n^k}(\gamma_n)\}_{n \in \mathbb{N}}$ for every $k \in \mathbb{N}$. Furthermore, the refutations are $\mathbf{P}$-uniform.*

We avoid going into any more details about $\mathbf{TC}^0$-Frege here, since it would require formally introducing threshold connectives into PK. We do remark, however, that from Corollary 6.8 one can show improved hardness of the Proof Analysis Problem (PAP) for $\mathbf{TC}^0$-Frege systems.

We recall the definition of the *Proof Analysis Problem* [AAdRK25, Definition 3.1]. Let Q be a propositional proof system. The *Proof Analysis Problem for Q* is the language

$$\mathrm{PAP}_Q := \{(\varphi, \pi, 1^s) \mid \varphi \in \mathrm{SAT} \text{ and } \pi : Q \vdash \neg \textsc{Ref}_s(\varphi)\}. \tag{6.33}$$

We denote by $\mathrm{PAP}_Q[s(n)]$ the problem where the size parameter s is restricted to be at least $s(n)$ for n the number of variables of φ. The problem asks, given the proof of a Resolution lower bound in a fixed proof system Q, to decide whether the underlying formula is satisfiable. We say that a propositional proof system Q is *analyzable* if there exists some constant $c > 0$ such that $\mathrm{PAP}_Q[n^c] \in \mathbf{P}$. If a propositional proof system $Q \geq \mathsf{Res}$ is both automatable and analyzable, then $\mathbf{P} = \mathbf{NP}$ [AAdRK25, Proposition 3.6]. As a result, showing that Q is hard to analyze suggests that, in order to show the hardness of automating Q, one needs techniques beyond the use of Ref formulas as in [AM20].

While it was known that analyzing Extended Frege systems is **NP**-hard under many-one polynomial-time reductions [AAdRK25, Corollary 5.6], for $\mathbf{TC}^0$-Frege systems the result was only known under quasipolynomial-time reductions. The fact that proving Resolution lower bounds within Q suffices for an **NP**-hardness reduction was shown in [AAdRK25, Corollary 5.7; AdRKR26, Theorem 8.4].

Corollary 6.9. *There exists a constant $d \in \mathbb{N}$ such that for every $c \in \mathbb{N}$, the problem $\mathrm{PAP}_{\mathbf{TC}^0_d\text{-Frege}}[n^c]$ is* **NP***-complete under polynomial-time many-one reductions.*

Acknowledgements

We are grateful to Susanna F. de Rezende for invaluable discussions on the ideas in this paper. We thank Ján Pich for drawing our attention to Razborov's reduction, as discussed in Section 1.2.1, which later also inspired the modifications for $\mathsf{Res}(k)$ in Section 3.2. We thank Albert Atserias for kind and helpful feedback and for pointing us to the conjecture about WPHP in his doctoral thesis (see Section 1.1.2). We thank Dimitrios Tsintsilidas for comments as well as for helpful suggestions towards an improved presentation of the results of this paper following the claimed refutation of the Exponential Time Hypothesis by OpenAI. We also thank Jonas Conneryd, Stefan Grosser, Svyatoslav Gryaznov, Ondřej Ježil, Erfan Khaniki, Jiaqi Lu, and Pavel Pudlák for other helpful comments and pointers.

Noel Arteche was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The author gratefully acknowledges that he has benefited greatly from being part of the Basic Algorithms Research Copenhagen (BARC) center financed by the Villum Investigator grant 54451. The author would also like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme *Logical Foundations of Complexity Theory*, where part of the work on this paper was undertaken; this work was supported by EPSRC grant EP/Z000580/1.

AI disclosure

The core ideas in this paper are due to the human author. This includes in particular the broad research questions, the ideas behind the new reduction in Section 3 and the formalization in Section 6. Large language models were used throughout the project for discussing these ideas further, testing parameters, performing literature reviews and sharpening the results. GPT 5.6-Sol and GPT 6 were used to generate initial drafts of some of the technical parts of the write-up based on the proofs of the author; in particular, the model generated the LaTeX code for the cedent calculus derivations in Section 3 and a large part of the formalization within bounded arithmetic in Section 6. GPT 6 was also used during the writing process to polish the writing, correct grammar, carry out technical proofreading and generate the TikZ code behind Figure 1 and Figure 2 based on hand-drawn

sketches by the human author. The author is responsible for the correctness of all mathematical arguments, citations, and claims included in the paper. The author thanks Lund University for providing access to these large language models.

References

[ADR26] N. Acheampong, B. Davis, and R. Robere, "Polynomial interpolation for Sherali–Adams lower bounds," Electronic Colloquium on Computational Complexity (ECCC), Tech. Rep. TR26-288, Oct. 2026.

[Ajt94] M. Ajtai, "The complexity of the pigeonhole principle," *Combinatorica*, vol. 14, no. 4, pp. 417–433, 1994.

[ABMP01] M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi, "Minimum propositional proof length is NP-hard to linearly approximate," *J. Symbolic Logic*, vol. 66, no. 1, pp. 171–191, 2001.

[AR08] M. Alekhnovich and A. A. Razborov, "Resolution is not automatizable unless **W**[P] is tractable," *SIAM Journal on Computing*, vol. 38, no. 4, pp. 1347–1363, 2008.

[AAdRK25] N. Arteche, A. Atserias, S. F. de Rezende, and E. Khaniki, "The proof analysis problem," in *2025 IEEE 66th Annual Symposium on Foundations of Computer Science (FOCS)*, Full version available at https://arxiv.org/abs/2506.16956, 2025, pp. 2558–2579.

[ACG24] N. Arteche, G. Carenini, and M. Gray, "Quantum automating $\mathbf{TC}^0$-Frege is LWE-hard," in *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, 2024, 15:1–15:25.

[AdRKR26] N. Arteche, S. F. de Rezende, E. Khaniki, and K. Risse, "The proof analysis problem for constant-depth Frege," in *2026 IEEE 67th Annual Symposium on Foundations of Computer Science (FOCS), to appear*, Full version available at https://arxiv.org/abs/2611.08305, 2026.

[Ats02] A. Atserias, "The complexity of resource-bounded propositional proofs," Ph.D. dissertation, Universitat Politècnica de Catalunya, Barcelona, Spain, Feb. 2002.

[Ats03] A. Atserias, "Improved bounds on the weak pigeonhole principle and infinitely many primes from weaker axioms," *Theoretical Computer Science*, vol. 295, no. 1–3, pp. 27–39, 2003.

[AB04] A. Atserias and M. L. Bonet, "On the automatizability of resolution and related propositional proof systems," *Inform. and Comput.*, vol. 189, no. 2, pp. 182–201, 2004.

[ABE02] A. Atserias, M. L. Bonet, and J. L. Esteban, "Lower bounds for the weak pigeonhole principle and random formulas beyond Resolution," *Information and Computation*, vol. 176, no. 2, pp. 136–152, 2002.

[AM20] A. Atserias and M. Müller, "Automating resolution is **NP**-hard," *Journal of the ACM (JACM)*, vol. 67, no. 5, pp. 1–17, 2020.

[BP96] P. Beame and T. Pitassi, "Simplified and improved resolution lower bounds," in *Proceedings of 37th Conference on Foundations of Computer Science*, 1996, pp. 274–282.

[BB19] A. Beckmann and S. Buss, "On transformations of constant depth propositional proofs," *Ann. Pure Appl. Logic*, vol. 170, no. 10, pp. 1176–1187, 2019.

[BB05] A. Beckmann and S. R. Buss, "Separation results for the size of constant-depth propositional proofs," *Ann. Pure Appl. Logic*, vol. 136, no. 1-2, pp. 30–55, 2005.

[BPT14] A. Beckmann, P. Pudlák, and N. Thapen, "Parity games and propositional proofs," *ACM Trans. Comput. Logic*, vol. 15, no. 2, May 2014.

[Bel20] Z. Bell, "Automating regular or ordered resolution is **NP**-hard," *Electronic Colloquium on Computational Complexity (ECCC)*, no. TR20-105, 2020.

[BW01] E. Ben-Sasson and A. Wigderson, "Short proofs are narrow—resolution made simple," *J. ACM*, vol. 48, no. 2, pp. 149–169, 2001.

[BDG+04] M. L. Bonet, C. Domingo, R. Gavaldà, A. Maciel, and T. Pitassi, "Non-automatizability of bounded-depth Frege proofs," *computational complexity*, vol. 13, pp. 47–68, 2004.

[BPR00] M. L. Bonet, T. Pitassi, and R. Raz, "On interpolation and automatization for Frege systems," *SIAM J. Comput.*, vol. 29, no. 6, pp. 1939–1967, 2000.

[BBP+05] J. Buresh-Oppenheim, P. Beame, T. Pitassi, R. Raz, and A. Sabharwal, "Bounded-depth Frege lower bounds for weaker pigeonhole principles," *SIAM Journal on Computing*, vol. 34, no. 2, pp. 261–276, 2005.

[BP98] S. Buss and T. Pitassi, "Resolution and the weak pigeonhole principle," in *Computer science logic (Aarhus, 1997)*, ser. Lecture Notes in Comput. Sci. Vol. 1414, Springer, Berlin, 1998, pp. 149–156, ISBN: 3-540-64570-5.

[BY24] S. Buss and E. Yolcu, "Regular resolution effectively simulates resolution," *Inform. Process. Lett.*, vol. 186, Paper No. 106489, 4, 2024.

[Bus86] S. R. Buss, *Bounded arithmetic*. Bibliopolis, Naples, 1986.

[Bus95] S. R. Buss, "On Gödel's theorems on lengths of proofs. II. Lower bounds for recognizing k symbol provability," in *Feasible mathematics, II (Ithaca, NY, 1992)*, ser. Progr. Comput. Sci. Appl. Logic, vol. 13, Birkhäuser Boston, Boston, MA, 1995, pp. 57–90, ISBN: 0-8176-3675-7.

[Bus04] S. R. Buss, "Bounded arithmetic and constant depth Frege proofs," in *Complexity of computations and proofs*, ser. Quad. Mat. Vol. 13, Dept. Math., Seconda Univ. Napoli, Caserta, 2004, pp. 153–174.

[CRVW02] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, "Randomness conductors and constant-degree lossless expanders," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC)*, 2002, pp. 659–668.

[CdR25] G. Carenini and S. F. de Rezende, "On the automatability of tree-like k-DNF resolution," in *40th Computational Complexity Conference*, ser. LIPIcs. Leibniz Int. Proc. Inform. Vol. 339, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2025, Art. No. 14, 21, ISBN: 978-3-95977-379-9.

[CLO24] L. Chen, J. Li, and I. C. Oliveira, "Reverse mathematics of complexity lower bounds," in *2024 IEEE 65th Annual Symposium on Foundations of Computer Science—FOCS 2024*, 2024, pp. 505–527.

[Cla26] Claude (Anthropic), *A $2^{o(n)}$ time algorithm for 3-SAT*, Preprint, 2026.

[CP90] S. Cook and T. Pitassi, "A feasibly constructive lower bound for resolution proofs," *Information Processing Letters*, vol. 34, no. 2, pp. 81–85, 1990.

[CR79] S. A. Cook and R. A. Reckhow, "The relative efficiency of propositional proof systems," *J. Symbolic Logic*, vol. 44, no. 1, pp. 36–50, 1979.

[CDL+16] M. Cygan et al., "On problems as hard as CNF-SAT," *ACM Transactions on Algorithms*, vol. 12, no. 3, 41:1–41:24, 2016.

[DHW06] E. Dantsin, E. A. Hirsch, and A. Wolpert, "Clause shortening combined with pruning yields a new upper bound for deterministic SAT algorithms," in *Algorithms and Complexity (CIAC 2006)*, ser. Lecture Notes in Computer Science, vol. 3998, Springer, 2006, pp. 60–68.

[DW05] E. Dantsin and A. Wolpert, "An improved upper bound for SAT," in *Theory and Applications of Satisfiability Testing (SAT 2005)*, Springer, 2005, pp. 400–407.

[DR26] B. Davis and R. Robere, "Res(log) proves bounded-depth frege lower bounds," *Electronic Colloquium on Computational Complexity (ECCC)*, no. TR26-055, 2026.

[dHKS15] R. de Haan, I. Kanj, and S. Szeider, "On the subexponential-time complexity of CSP," *Journal of Artificial Intelligence Research*, vol. 52, pp. 203–234, 2015.

[dRez21] S. F. de Rezende, "Automating Tree-like Resolution in time $n^{o(\log n)}$ is ETH-hard," *Procedia Computer Science*, vol. 195, pp. 152–162, 2021, Proceedings of the XI Latin and American Algorithms, Graphs and Optimization Symposium.

[dRFJ+25] S. F. de Rezende, N. Fleming, D. A. Janett, J. Nordström, and S. Pang, "Truly super-critical trade-offs for resolution, cutting planes, monotone circuits, and Weisfeiler–Leman," in *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC)*, 2025, pp. 1371–1382.

[dRGN+21] S. F. de Rezende, M. Göös, J. Nordström, T. Pitassi, R. Robere, and D. Sokolov, "Automating algebraic proof systems is **NP**-hard," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021, pp. 209–222.

[GdHKS16] R. Ganian, R. de Haan, I. Kanj, and S. Szeider, "On existential MSO and its relation to ETH," in *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, ser. LIPIcs, vol. 58, 2016, 42:1–42:14.

[Gar19] M. Garlík, "Resolution lower bounds for refutation statements," in *44th International Symposium on Mathematical Foundations of Computer Science*, 2019, Art. No. 37, 13.

[Gar24] M. Garlík, "Failure of feasible disjunction property for k-DNF resolution and **NP**-hardness of automating it," in *39th Computational Complexity Conference, CCC 2024, July 22-25, 2024, Ann Arbor, MI, USA*, vol. 300, 2024, 33:1–33:23.

[GGL+26] M. Garlík, S. Gryaznov, J. Lu, R. Santhanam, and I. Tzameret, "Meta-mathematics of algebraic complexity," in *41st Annual Symposium on Logic in Computer Science (LICS 2026)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 380, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026, 49:1–49:25.

[GGRT26] M. Garlík, S. Gryaznov, H. Ren, and I. Tzameret, "The weak rank principle: Lower bounds and applications," in *Proceedings of the 58th Annual ACM Symposium on Theory of Computing (STOC)*, ACM, 2026, pp. 138–149.

[GKMP20] M. Göös, S. Koroth, I. Mertz, and T. Pitassi, "Automating cutting planes is **NP**-hard," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 2020, pp. 68–77.

[GMRS25] M. Göös, G. Maystre, K. Risse, and D. Sokolov, "Supercritical tradeoffs for monotone circuits," in *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC)*, 2025. arXiv: 2411.14268.

[HP93] P. Hájek and P. Pudlák, *Metamathematics of first-order arithmetic*. Springer-Verlag, Berlin, 1993.

[Hak85] A. Haken, "The intractability of resolution," *Theoret. Comput. Sci.*, vol. 39, no. 2-3, pp. 297–308, 1985.

[Hås21] J. Håstad, "On small-depth Frege proofs for Tseitin for grids," *Journal of the ACM*, vol. 68, no. 1, 1:1–1:31, 2021.

[Hås25] J. Håstad, "On small-depth Frege proofs for PHP," *TheoretiCS*, vol. 4, 2025.

[HR25] J. Håstad and K. Risse, "On bounded depth proofs for Tseitin formulas on the grid; revisited," *SIAM Journal on Computing*, vol. 54, no. 5, FOCS22-288–FOCS22-339, 2025.

[IP01] R. Impagliazzo and R. Paturi, "On the complexity of k-SAT," *Journal of Computer and System Sciences*, vol. 62, no. 2, pp. 367–375, 2001.

[IPZ01] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?" *Journal of Computer and System Sciences*, vol. 63, no. 4, pp. 512–530, 2001.

[IR22] D. Itsykson and A. Riazanov, "Automating OBDD proofs is **NP**-hard," in *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, vol. 241, 2022, 59:1–59:15.

[Iwa97] K. Iwama, "Complexity of finding short resolution proofs," in *Mathematical foundations of computer science 1997 (Bratislava)*, ser. Lecture Notes in Comput. Sci. Vol. 1295, Springer, Berlin, 1997, pp. 309–318.

[JT26] O. Ježil and D. Tsintsilidas, *Private communication*, 2026.

[Kra94] J. Krajíček, "Lower bounds to the size of constant-depth propositional proofs," *Journal of Symbolic Logic*, vol. 59, no. 1, pp. 73–86, 1994.

[Kra95] J. Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*. Cambridge University Press, Cambridge, 1995.

[Kra01] J. Krajíček, "On the weak pigeonhole principle," *Fundamenta Mathematicae*, vol. 170, no. 1-2, pp. 123–140, 2001.

[Kra19] J. Krajíček, *Proof complexity*. Cambridge University Press, Cambridge, 2019.

[KP98] J. Krajíček and P. Pudlák, "Some consequences of cryptographical conjectures for S^1_2 and EF," *Information and Computation*, vol. 140, no. 1, pp. 82–94, 1998.

[KPW95] J. Krajíček, P. Pudlák, and A. Woods, "An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle," *Random Structures Algorithms*, vol. 7, no. 1, pp. 15–39, 1995.

[LLR26] J. Li, Y. Li, and H. Ren, "Finding bugs in short proofs: The metamathematics of resolution lower bounds," in *STOC'26—Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, ACM, New York, 2026, pp. 788–798, ISBN: 979-8-4007-2536-4.

[MPW02] A. Maciel, T. Pitassi, and A. R. Woods, "A new proof of the weak pigeonhole principle," *J. Comput. System Sci.*, vol. 64, no. 4, pp. 843–872, 2002, Special issue on STOC 2000 (Portland, OR).

[MPW19] I. Mertz, T. Pitassi, and Y. Wei, "Short proofs are hard to find," in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, vol. 132, 2019, 84:1–84:16.

[ODo26] R. O'Donnell, *Some exposition of "the exponential time hypothesis is false"*, Expository manuscript, Oct. 2026.

[Ope26] OpenAI, *The exponential time hypothesis is false*, Preprint, 2026.

[Pap24] T. Papamakarios, "Depth-d Frege systems are not automatable unless **P** = **NP**," in *39th Computational Complexity Conference (CCC 2024)*, vol. 300, 2024, 22:1–22:17.

[PW85] J. Paris and A. Wilkie, "Counting problems in bounded arithmetic," in *Methods in mathematical logic (Caracas, 1983)*, ser. Lecture Notes in Math. Vol. 1130, Springer, Berlin, 1985, pp. 317–340.

[PWW88] J. B. Paris, A. J. Wilkie, and A. R. Woods, "Provability of the pigeonhole principle and the existence of infinitely many primes," *J. Symbolic Logic*, vol. 53, no. 4, pp. 1235–1244, 1988, ISSN: 0022-4812,1943-5886.

[PW10] M. Pătraşcu and R. Williams, "On the possibility of faster SAT algorithms," in *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010, pp. 1065–1075.

[PBI93] T. Pitassi, P. Beame, and R. Impagliazzo, "Exponential lower bounds for the pigeonhole principle," *Computational Complexity*, vol. 3, no. 2, pp. 97–140, 1993.

[PRST16] T. Pitassi, B. Rossman, R. A. Servedio, and L.-Y. Tan, "Poly-logarithmic Frege depth lower bounds via an expander switching lemma," in *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC)*, ACM, 2016, pp. 644–657.

[Pud03] P. Pudlák, "On reducibility and symmetry of disjoint **NP** pairs," *Theoretical Computer Science*, vol. 295, no. 1-3, pp. 323–339, 2003.

[Pud20] P. Pudlák, *Reflection principles, propositional proof systems, and theories*, 2020. arXiv: 2007.14835.

[Raz04a] R. Raz, "Resolution lower bounds for the weak pigeonhole principle," *Journal of the ACM*, vol. 51, no. 2, pp. 115–138, 2004.

[Raz95] A. A. Razborov, "Bounded arithmetic and lower bounds in Boolean complexity," in *Feasible mathematics, II (Ithaca, NY, 1992)*, ser. Progr. Comput. Sci. Appl. Logic, vol. 13, Birkhäuser Boston, Boston, MA, 1995, pp. 344–386.

[Raz98] A. A. Razborov, "Lower bounds for the polynomial calculus," *Computational Complexity*, vol. 7, no. 4, pp. 291–324, 1998.

[Raz02] A. A. Razborov, "Proof complexity of pigeonhole principles," in *Developments in Language Theory*, ser. Lecture Notes in Computer Science, vol. 2295, Springer, 2002, pp. 100–116.

[Raz04b] A. A. Razborov, "Resolution lower bounds for perfect matching principles," *J. Comput. System Sci.*, vol. 69, no. 1, pp. 3–27, 2004.

[Raz15] A. A. Razborov, "Pseudorandom generators hard for k-DNF Resolution and polynomial calculus resolution," *Annals of Mathematics*, vol. 181, no. 2, pp. 415–472, 2015.

[SS12] R. Santhanam and S. Srinivasan, "On the limits of sparsification," in *Automata, Languages, and Programming (ICALP 2012), Part I*, ser. Lecture Notes in Computer Science, vol. 7391, Springer, 2012, pp. 774–785.

[ST21] R. Santhanam and I. Tzameret, "Iterated lower bound formulas: A diagonalization-based approach to proof complexity," in *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, 2021, pp. 234–247.

[Sch05] R. Schuler, "An algorithm for the satisfiability problem of formulas in conjunctive normal form," *Journal of Algorithms*, vol. 54, no. 1, pp. 40–44, 2005.

[SBI04] N. Segerlind, S. R. Buss, and R. Impagliazzo, "A switching lemma for small restrictions and lower bounds for k-DNF Resolution," *SIAM Journal on Computing*, vol. 33, no. 5, pp. 1171–1200, 2004.

[Sub26] B. Subercaseaux. "Observation on the subexponential-time simulation of nondeterministic computation." Post on X (formerly Twitter). [Online]. Available: https://x.com/b_subercaseaux/status/2114865520055046442

[Urq87] A. Urquhart, "Hard examples for resolution," *Journal of the ACM*, vol. 34, no. 1, pp. 209–219, 1987.

[Wil78] A. J. Wilkie, "Some results and problems on weak systems of arithmetic," in *Logic Colloquium '77 (Proc. Conf., Wrocław, 1977)*, ser. Stud. Logic Found. Math. Vol. 96, North-Holland, Amsterdam-New York, 1978, pp. 285–296.

A Related results on the complexity of pigeonhole principles

Our understanding of the complexity of WPHP within constant-depth Frege is partial at best. The only situation where we have almost full grasp of how the stretch affects the size of the proofs is in Resolution (depth-0 Frege), where we have lower bounds in essentially all regimes. Razborov's survey [Raz02] is still a good reference for this. Table 1 summarizes the main known upper and lower bounds.

The case of $\mathsf{Res}(k)$ For $\mathsf{Res}(k)$, the situation is already much more unclear. Atserias, Bonet, and Esteban [ABE02] proved that $\mathsf{Res}(2)$ requires exponential size to refute PHP_n^{2n}. For constant k, the setting considered in our paper, Segerlind, Buss, and Impagliazzo [SBI04] extended exponential lower bounds to $\mathsf{Res}(k)$, even allowing k to grow as $\sqrt{\log n/\log\log n}$. Razborov [Raz15, Theorem 2.15] subsequently strengthened the bound to $\exp\big(n/(\log n)^{O(k)}\big)$, which remains exponential for $k \leq \varepsilon \log n/\log\log n$ and a sufficiently small constant $\varepsilon > 0$. His result also holds for the onto version of the principle. Despite these results for PHP_n^{2n}, the complexity of $\mathrm{PHP}_n^{n^2}$ even as low as $\mathsf{Res}(2)$ is entirely open. Our Hypothesis 1.9 asks for superpolynomial hardness in the more extreme regime where the stretch is exponentially large, in analogy to the situation in Resolution, where Razborov's lower bound of $\exp\big(\Omega(n^{1/3})\big)$ holds for the onto weak pigeonhole principle with arbitrarily many pigeons [Raz04b]. It appears perfectly plausible that the same situation as with Resolution holds for $\mathsf{Res}(k)$, hence motivating Hypothesis 1.9, but an alternative outcome would be equally exciting.

The case of bounded-depth Frege Building on Paris, Wilkie, and Woods [PWW88] and Krajíček [Kra95], Maciel, Pitassi, and Woods obtained depth-0.5 proofs of size $n^{O(\log n)}$ for $\mathrm{PHP}_n^{n^2}$ and size $n^{O((\log n)^2)}$ for PHP_n^{2n}. Allowing greater constant depth improves the size further: Atserias [Ats03] proved that, for every fixed integer $d \geq 2$, PHP_n^{2n} has proofs of depth $O(d)$ and size $n^{O_d((\log n)^{2/d})}$. For $\mathrm{PHP}_n^{n^2}$, the original bound $n^{O_d(\log^{(d)} n)}$ of Paris, Wilkie and Woods [PWW88] is better than the one by Atserias. Observe as well that for every $\varepsilon > 0$, some fixed depth admits proofs of size $\exp\big(O((\log n)^{1+\varepsilon})\big)$ of $\mathrm{PHP}_n^{n^2}$. The depth depends on ε and these results do not yield polynomial-size proofs at any fixed depth.

Proof system	Pigeons (m)	Size bound	References
Lower bounds			
Resolution	$n+1$	$\exp(\Omega(n))$	[Hak85]
Resolution	Any $m > n$	$\exp\left(\Omega(n^{1/3})\right)$	[Raz04b]
Res(k)	$2n$	$\exp\left(n/(\log n)^{O(k)}\right)$	[Raz15]
Depth-d Frege	$n+1$	$\exp\left(n^{\Omega_d(1)}\right)$	[PBI93; KPW95]
Depth-d Frege	$n < m \le n + \frac{n}{(\log n)^c}$	$2^{(\log n)^a}$	[BBP+05]
Upper bounds			
Resolution	Any $m > n$	$\exp(O(n))$	[BP98]
Resolution	$m \ge \left\lceil 2^{\sqrt{n \log n}} \right\rceil$	$m^{O(1)}$	[BP98]
Depth-0.5 Frege	$2n$	$n^{O((\log n)^2)}$	[MPW02]
Depth-0.5 Frege	n^2	$n^{O(\log n)}$	[MPW02]
Depth-1.5 Frege	$2n$	$n^{O(\log n)}$	[PWW88; Kra94]
Depth-$O(d)$ Frege	$2n$	$n^{O_d((\log n)^{2/d})}$	[Ats03]
Depth-$O(d)$ Frege	n^2	$n^{O_d(\log^{(d)} n)}$	[PWW88]
Depth-$O(1/\varepsilon)$ Frege	$m \ge 2^{n^\varepsilon}$ for $\varepsilon > 0$	$m^{O(\varepsilon)}$	[PW85]

Table 1: Summary of known size upper and lower bounds for refutations of the unary pigeonhole principle PHP_n^m with m pigeons and n holes. Depth is measured using the convention of this paper. In the near-linear-stretch lower bound of [BBP+05], for every fixed d and every $a > 0$, there exists $c = c(d, a) > 0$ for which the bound applies. In the upper bounds, the depths $d \ge 2$ for [Ats03] and $d \ge 1$ for [PWW88] are fixed, and $\log^{(d)}$ denotes the d-fold iterated logarithm. The Res(k) lower bound applies also to the onto version and is $\exp\left(n^{\Omega(1)}\right)$ for $k \le \varepsilon \log n / \log\log n$ and sufficiently small absolute $\varepsilon > 0$.

The only known bounded-depth Frege lower bounds concern much smaller stretch. Ajtai proved superpolynomial lower bounds for PHP_n^{n+1} [Ajt94], subsequently strengthened to $\exp\left(n^{\Omega_d(1)}\right)$ at every fixed depth d by Pitassi, Beame, and Impagliazzo and independently by Krajíček, Pudlák, and Woods [PBI93; KPW95]. Buresh-Oppenheim, Beame, Pitassi, Raz, and Sabharwal [BBP+05] extended this range: for every fixed depth d and every $a > 0$, there is a constant $c = c(d, a)$ such that if $m \le n + \frac{n}{(\log n)^c}$, then depth-$d$ proofs require size at least $2^{(\log n)^a}$. The dependence of c on both d and a is essential. These results do not reach $m = 2n$, let alone $m = n^2$.

B Layered Resolution refutations

In this appendix we briefly show that any Resolution refutation can be made *layered* with a small blow-up in size, meaning that a derivation at depth k only uses premises at depth $k+1$.

Proposition B.1. *Let φ be an unsatisfiable CNF formula admitting a Resolution refutation π of length L and depth $D \ge 1$. There exists a Resolution refutation π' of φ of length $O(LD)$ and depth D, such that every root-to-leaf path in its refutation DAG has length exactly D. In particular, π' is a layered Resolution refutation.*

Proof. We construct π' by inserting trivial weakening steps into π. We measure depth from the root

$\bot$, which has depth 0. For each vertex v of the refutation DAG, let $d(v)$ be the length of the longest directed path from the root to v. Thus, $d(\bot) = 0$, and every edge $v \to u$ satisfies $d(u) \geq d(v)+1$. First, for every edge $v \to u$ with $d(u) > d(v) + 1$, subdivide the edge by inserting $d(u) - d(v) - 1$ copies of the clause labeling u. These copies are obtained by identity weakenings $C \vdash C$. Consequently, every edge now connects consecutive layers. Next, for every leaf at depth $k < D$, replace it by a chain of $D - k$ identity weakenings, starting from the corresponding initial clause at depth D.

The resulting DAG is a valid Resolution refutation in which every edge connects consecutive layers and every leaf lies at depth D. Hence every root-to-leaf path has length exactly D. Finally, the original DAG has L vertices and at most $2L$ edges. Each edge introduces at most D new vertices, and padding the leaves introduces at most LD further vertices. Therefore, $\text{length}(\pi') = O(LD)$. The depth remains D, and it is easy to see that as a consequence of this transformation, the refutation is layered. □

C Stretch amplification

In this appendix we prove Proposition 4.1. This shows that pigeonhole principles with polynomial stretch are inter-derivable within constant-depth Frege with at most a constant increase in the depth of the proofs.

Proof of Proposition 4.1. We prove the proposition for the special case $\eta = 1$, which already contains all the ideas needed for the general case. We distinguish two cases. First, if $0 < \delta \leq 1$, since $\lceil n^{1+\delta} \rceil \leq n^2$, every refutation of $\text{PHP}_n^{\lceil n^{1+\delta} \rceil}$ is also a refutation of $\text{PHP}_n^{n^2}$, after identifying the former with the subformula obtained by restricting to an arbitrary set of $\lceil n^{1+\delta} \rceil$ pigeons. Thus the claim holds without any increase in depth or size.

If $\delta > 1$, then fix $r := \lceil 1 + \delta \rceil$. We show that a refutation of $\text{PHP}_n^{\lceil n^{1+\delta} \rceil}$ yields, with polynomial overhead and a constant increase in depth, a refutation of $\text{PHP}_n^{n^2}$. Let the variables of $\text{PHP}_n^{n^2}$ be written as $p_{(x_1,x_2),y}$, for $x_1, x_2, y \in [n]$. These variables encode an injective relation from $[n]^2$ to $[n]$. By composing this relation with itself $r - 1$ times, we obtain formulas $Q^{(r)}_{\bar{x},y}$, for $\bar{x} \in [n]^r$ and $y \in [n]$, encoding an injective relation from $[n]^r$ to $[n]$.

More precisely, set

$$Q^{(1)}_{x,y} := \begin{cases} \top & \text{if } x = y, \\ \bot & \text{otherwise,} \end{cases} \tag{C.1}$$

and recursively define

$$Q^{(j+1)}_{(x_1,\ldots,x_{j+1}),y} := \bigvee_{z \in [n]} \left(Q^{(j)}_{(x_1,\ldots,x_j),z} \wedge p_{(z,x_{j+1}),y} \right). \tag{C.2}$$

Since r is constant, these formulas have constant depth and size $n^{O_\delta(1)}$.

We claim that, for every $j \leq r$, the axioms of $\text{PHP}_n^{n^2}$ admit polynomial-size constant-depth derivations of $\bigvee_{y \in [n]} Q^{(j)}_{\bar{x},y}$ for every $\bar{x} \in [n]^j$ and $\neg Q^{(j)}_{\bar{x},y} \vee \neg Q^{(j)}_{\bar{x}',y}$, for every pair of distinct tuples $\bar{x}, \bar{x}' \in [n]^j$ and every $y \in [n]$.

This follows by induction on j. The base case is immediate from the definition of $Q^{(1)}$. For the induction step, totality follows by combining the totality of $Q^{(j)}$ with the pigeon axioms for $p_{(z,x_{j+1}),y}$.

For injectivity, suppose two distinct tuples $\bar{x}, \bar{x}' \in [n]^{j+1}$ have the same image. Expanding their definitions, there must be $z, z' \in [n]$ such that $p_{(z,x_{j+1}),y} \wedge p_{(z',x'_{j+1}),y}$ holds. The injectivity axioms

of $\mathrm{PHP}_n^{n^2}$ force $z = z'$ and $x_{j+1} = x'_{j+1}$. The induction hypothesis applied to $Q^{(j)}$ then forces the remaining coordinates to agree, contradicting $\bar{x} \neq \bar{x}'$. These arguments can be implemented by expanding the definitions and applying the rules of PK. Since $j \leq r$ is constant, the resulting derivations have size $n^{O_\delta(1)}$ and constant depth.

Now choose $\lceil n^{1+\delta} \rceil$ distinct tuples $\bar{x}_1, \ldots, \bar{x}_{\lceil n^{1+\delta} \rceil} \in [n]^r$ and substitute $p_{i,y} \longmapsto Q^{(r)}_{\bar{x}_i, y}$ in the assumed PK_d-refutation of $\mathrm{PHP}_n^{\lceil n^{1+\delta} \rceil}$. The preceding derivations establish every substituted pigeon and injectivity axiom from $\mathrm{PHP}_n^{n^2}$.

Since the substituted formulas have constant depth and polynomial size, the resulting refutation has depth at most $d + c_\delta$ for some constant c_δ depending only on δ, and total size $T \cdot n^{O_\delta(1)}$. This completes the proof for $\eta = 1$.

The generalization for all $\eta > 0$ is straightforward. Indeed, if $\eta \geq \delta$, the result follows immediately by weakening. Otherwise, the same composition argument outlined above applies with $[n]^2$ replaced by $[n] \times [a]$, where $a = \lfloor n^{1+\eta}/n \rfloor$. Iterating the construction a constant number of times produces $na^r \geq \lceil n^{1+\delta} \rceil$ pigeons, provided $r\eta > \delta$. The bounds on depth and size remain unchanged up to constants depending on δ and η. □