Automating Resolution is NP-Hard

Albert Atserias and Moritz Müller
Universitat Politècnica de Catalunya

September 10, 2019

Abstract

We show that the problem of finding a Resolution refutation that is at most polynomially longer than a shortest one is NP-hard. In the parlance of proof complexity, Resolution is not automatizable unless P = NP. Indeed, we show that it is NP-hard to distinguish between formulas that have Resolution refutations of polynomial length and those that do not have subexponential length refutations. This also implies that Resolution is not automatizable in subexponential time or quasi-polynomial time unless NP is included in SUBEXP or QP, respectively.

1 Introduction

The proof search problem for a given proof system asks, given a tautology, to find an approximately shortest proof of it. Clearly, the computational complexity of such problems is of fundamental importance for automated theorem proving. In particular, among the proof systems for propositional logic, Resolution deserves special attention since most modern implementations of satisfiability solvers are based on it.

We say that the proof search problem for Resolution is solvable in polynomial time if there is an algorithm that, given a contradictory CNF formula F as input, outputs a Resolution refutation of F in time polynomial in $r + s$, where r is the size of F, and s is the length of a shortest Resolution refutation of F. More succinctly, we say that Resolution is *automatizable* [11]. It is clear that the concept of automatizability applies not only to Resolution but to any refutation or proof system, and one can ask for automating algorithms that run in quasi-polynomial time, subexponential time, etc..[1]

In this paper we show that Resolution is not automatizable unless P = NP. The assumption is clearly optimal since P = NP implies that it is. To prove our result we give a direct and efficient reduction from 3-SAT, the satisfiability problem for 3-CNF formulas. The reduction is so efficient that it also rules out quasi-polynomial and subexponential time

[1] The time of the automating algorithm is not measured in r but in $r + s$ because s can be much larger than r. We use both r and s, and not just s, because a Resolution refutation need not use all clauses in F, but the algorithm should be given the opportunity to at least read all of F.

automating algorithms for Resolution under the corresponding hardness assumptions. More precisely, let QP and SUBEXP denote the classes of problems that are decidable in quasi-polynomial time $2^{(\log n)^{O(1)}}$, and in subexponential time $2^{n^{o(1)}}$, respectively. Then our main result reads:

Theorem 1.

1. *Resolution is not automatizable in subexponential time unless* $\mathrm{NP} \subseteq \mathrm{SUBEXP}$.
2. *Resolution is not automatizable in quasi-polynomial time unless* $\mathrm{NP} \subseteq \mathrm{QP}$.
3. *Resolution is not automatizable in polynomial time unless* $\mathrm{NP} \subseteq \mathrm{P}$.

That Resolution is not automatizable in polynomial time has been known under a stronger assumption from parameterized complexity theory, using a more contrived reduction [1]: we review the literature below. The first two statements in Theorem 1 give the first evidence that Resolution is not automatizable in quasi-polynomial or subexponential time. As in the third statement, their assumptions are also optimal in that $\mathrm{NP} \subseteq \mathrm{QP}$ and $\mathrm{NP} \subseteq \mathrm{SUBEXP}$ imply that Resolution can be automated in quasi-polynomial and subexponential time, respectively.

The main result as stated in Theorem 1 is a direct consequence of the fact, which we also prove, that the problem of non-trivially approximating minimum proof length for Resolution is NP-hard. If for a CNF formula G we write $r(G)$ for the size of G, and $s(G)$ for the length of a shortest Resolution refutation of G, then we show:

Theorem 2. *There are reals $c > 0$ and $d > 0$ and a polynomial-time computable function G that maps any 3-CNF formula F to a CNF formula $G(F)$ such that, for $r = r(G(F))$ and $s = s(G(F))$:*

(a) if F is satisfiable, then $s < r^c$;

(b) if F is unsatisfiable, then $s > 2^{r^{1/d}}$.

Moreover, c and d can be chosen arbitrarily close to 1 and 2, respectively, which means that it is NP-hard to approximate the minimal Resolution refutation length to within $2^{r^{1/2-\epsilon}}$ for any $\epsilon > 0$.

Proof idea An idea of how a map G as in Theorem 2 could be defined is implicit in [36]. Pudlák [36, Theorem 2] maps a formula F to $\mathrm{REF}(F, s)$, for some s suitable for his context, where $\mathrm{REF}(F, s)$ is a CNF formula whose clauses describe, in a natural way, the Resolution refutations of F of length s. He used this function to show that the canonical pair of Resolution is symmetric. In particular, he showed that, if F is satisfiable, then $\mathrm{REF}(F, s)$ has a short Resolution refutation. This refutation proceeds naturally by using a satisfying assignment for F as a guide to find a true literal in each line of the alleged refutation, line by line one after another, until it gets stuck at the final empty clause. Conversely, we would like to show that, if F is unsatisfiable, then $\mathrm{REF}(F, s)$ is hard for Resolution. Intuitively, this should be the case: refuting $\mathrm{REF}(F, s)$ means proving a lower bound and "our experience rather suggests that proving lower bounds is difficult" – this is what Pudlák [36, Section 3] states about a similar formula for strong proof systems.

However, even after considerable time and effort, we failed to prove a Resolution length lower bound for $\mathrm{REF}(F, s)$. We bypass the issue by considering a *harder* version $\mathrm{RREF}(F, s)$ of $\mathrm{REF}(F, s)$. The harder $\mathrm{RREF}(F, s)$ is obtained from *relativizing* $\mathrm{REF}(F, s)$ seen as the propositional encoding of a first-order formula with a built-in linear order, following the general relativization technique of Dantchev and Riis [19]. When F is satisfiable, Pudlák's upper bound for $\mathrm{REF}(F, s)$ goes through to $\mathrm{RREF}(F, s)$, and the linear order is crucial in this. On the other hand, a random restriction argument in the style of [19] reduces a length lower bound for $\mathrm{RREF}(F, s)$ to a certain width lower bound for $\mathrm{REF}(F, s)$. The bulk of the current work is in establishing this width lower bound for $\mathrm{REF}(F, s)$, when F is unsatisfiable. It is proved by showing that, even if $s = s(n)$ has (not too slow) polynomial growth, the formulas $\mathrm{REF}(F, s)$ and $\mathrm{REF}(F, 2^{n+1})$ are indistinguishable by inferences of bounded width, where n is the number of variables in F. Since every unsatisfiable CNF formula with n variables has a refutation of length 2^{n+1}, the formula $\mathrm{REF}(F, 2^{n+1})$ is satisfiable, from which it follows that $\mathrm{REF}(F, s)$ does not have bounded-width refutations.

The technical device that we use in the indistinguishability argument is a variant of the *conditions* from [34], a particular formalization of a Prover-Adversary argument as, e.g., in [19]. The wording is meant to point out some analogy with forcing conditions [6]. This is not straightforward. The main obstacle overcome by our variant is the presence of the built-in linear order in $\mathrm{REF}(F, s)$. In fact, Dantchev and Riis [19, Section 5] point out explicitly that their arguments fail in the presence of a built-in linear order.

History of the problem The complexity of the proof search problem has been extensively investigated. Krajíček and Pudlák [33] showed that Extended Frege systems[2] are not automatizable assuming RSA is secure against P/poly. Subsequently, Bonet et al. showed this for Frege [11] and bounded depth Frege systems [12] assuming the Diffie-Hellman key exchange is secure against polynomial or, respectively, subexponential size circuits.

In fact, these results rule out feasible interpolation, an influential concept introduced to proof complexity by Krajíček [27, 29]. We refer to [32, Chapters 17, 18] for an account. If a system with feasible interpolation has short refutations of the contradictions that state that a pair of NP problems are not disjoint, then the pair can be separated by small circuits. Hence, feasible interpolation can be ruled out by finding short proofs of the disjointness of an NP pair that is hard to separate. Such hardness assumptions turn up naturally in cryptography [23] which explains the type of assumptions that were used in the results above.

The failure of feasible interpolation for a natural system R implies (cf. [4, Theorem 3]) that R is not even *weakly* automatizable in the sense that it would be polynomially simulated (see [18]) by an automatizable system. Hence, the above results left open whether weak proof systems, in particular those having feasible interpolation such as Resolution [29], were (weakly) automatizable. We refer to [3] for a survey, and focus from now on on Resolution.

[2] We refer to the textbook [28, Chapter 4] for a definition of this and the following systems. All notions relevant to state and prove our results are going to be defined later.

Pudlák showed [36, Corollary 2] that the weak automatizability of a proof system is equivalent to the (polynomial time) separability of its, so-called, canonical NP-pair [37]. This is, informally, the feasibility of distinguishing between satisfiable formulas and those with short refutations. Hence, to rule it out it suffices to reduce some inseparable disjoint NP pair to it. Atserias and Maneva [5] found in this respect useful pairs associated to two player games. The two NP sets collect the games won by the respective players, and separation means deciding the game. Following [5, 24], Beckmann et al. [9] showed that Resolution is not weakly automatizable unless parity games are decidable in polynomial time. Note, however, that this might well be the case, in fact, parity games are decidable in quasi-polynomial time [14].

Moreover, some non-trivial automating algorithms are known. Beame and Pitassi [8] observed that treelike Resolution *is* automatizable in quasi-polynomial time. For general Resolution there is an algorithm that, when given a 3-CNF formula with n variables that has a Resolution refutation of length at most s, computes a refutation in time $n^{O(\sqrt{n \log s})}$. This follows from the size-width trade-off of Ben-Sasson and Wigderson [10]. Indeed, it is trivial to find a refutation of width at most w in time $n^{O(w)}$ if there is one (and, in general, time $n^{\Omega(w)}$ is necessary [7]). When s is subexponential the runtime of this algorithm is the non-trivial $2^{n^{1/2+o(1)}}$.

However, the automatizability of Resolution is unlikely. First, Alekhnovich et al. [2] showed, assuming only P $\neq$ NP, that automatization is not possible in linear time. In fact, they proved more. They considered the optimization problem of finding, given a contradictory CNF, a Resolution refutation that is as short as possible. They reduced to it the optimization problem MMCSA of finding, given a monotone circuit, a satisfying assignment that has Hamming weight as small as possible. Known PCP theorems imply that this problem is not approximable with superconstant but sublinear ratio $2^{\log^{1-o(1)} n}$, so the same holds for finding short Resolution refutations. This argument can be adapted to many other refutation systems (see [2]).

But the main convincing evidence that Resolution is not automatizable, before the result of this paper, was achieved by Alekhnovich and Razborov [1]. By a different and ingenious reduction they showed that if Resolution, or even treelike Resolution, were automatizable, then MMCSA would have, in the terminology of parameterized complexity theory (see [16, Proposition 5]), an fpt algorithm with constant approximation ratio. Now, the same paper [1] also established "the first nontrivial parameterized inapproximability result" [20, p.9] by further deriving a randomized fpt algorithm for the parameterized decision version of MMCSA, a well-known W[P]-complete problem (see e.g. [21, Theorem 3.14]). The randomized fpt algorithm has subsequently been derandomized by Eickmeyer et al. [20], hence Resolution is not automatizable unless W[P] = FPT. Very recently, Mertz et al. [26] showed that Resolution is not automatizable in time $n^{(\log \log n)^{0.14}}$ unless ETH fails; this follows the same line of argument as [1] but is based on a more recent parameterized inapproximability result due to Chen and Lin [17].

Since these results apply not only to Resolution but even to treelike Resolution, which is automatizable in quasipolynomial time, Alekhnovich and Razborov stated that the "main problem left open" [1, Section 5] is whether general Resolution is automatizable in quasi-polynomial time. We consider Theorem 1 as an answer to this question.

The computational problem of computing minimal proof lengths also has a long history. For first-order logic, the problem dates back to Gödel's famous letter to von Neumann; we refer to [35] for a historical discussion, to [13] for a proof of Gödel's claim in the letter, and to [15] for some more recent results. In propositional logic, the problem has been shown to be NP-hard for a particular Frege system by Buss [13], and for Resolution by Iwama [25]. Alekhnovich et al. [2] showed that the minimal Resolution refutation length cannot be approximated to within any fixed polynomial unless NP $\not\subseteq$ P/poly: for every $d \in \mathbb{N}$ there are functions G and S, computable in non-uniform polynomial time, such that for every CNF formula F of sufficiently large size $r = r(F)$ we have either $s(G(F)) < S(r)$ or $s(G(F)) > S(r)^d$ depending on the satisfiability F. This falls short to rule out automatizability because $S(r)$ has exponential growth. Earlier, Iwama [25] found uniformly computable such functions with polynomially bounded $S(r)$ but his gap was only $S(r)$ versus $S(r) + r^d$ for a constant d, so also falls short to rule out automatizability.

Outline In Section 2 we introduce some notation and basic terminology from propositional logic. Section 3 presents Resolution refutations as finite structures. Section 4 is devoted to $\mathrm{REF}(F, s)$ and proves the width lower bound when F is unsatisfiable (Lemma 4). Section 5 discusses the relativized formula $\mathrm{RREF}(F, s)$, the refutation length upper bound when F is satisfiable (Lemma 11), and the refutation length lower bound when F is unsatisfiable (Lemma 10). Theorems 2 and 1 are derived from these lemmas in Section 6. In Section 7 we discuss some open issues. Finally, for easiness of reference, in Appendix A we give the detailed lists of clauses for the formulas REF and RREF.

2 Preliminaries

For $n \in \mathbb{N}$ we let $[n] := \{1, \dots, n\}$ and understand that $[0] = \emptyset$. A *partial function* from a set A to a set B is a function f with domain $\mathrm{Dom}(f)$ included in A and image $\mathrm{Img}(f)$ included in B. We view partial functions from A to B as sets of ordered pairs $(u, v) \in A \times B$. For any set C, the *restriction of f to C* is $f \cap (C \times \mathrm{Img}(f))$. The *restriction of f with image C* is $f \cap (\mathrm{Dom}(f) \times C)$.

We fix some notation for propositional logic. Let V be a set of propositional variables that take truth values in $\mathrm{B} = \{0, 1\}$, where 0 denotes *false* and 1 denotes *true*. A *literal* is a variable X or its negation $\neg X$, also denoted $\bar{X}$. We also write $X^{(1)}$ for X and $X^{(0)}$ for $\bar{X}$. A *clause* is a set of literals, that we write as a disjunction of its elements. A clause is *non-tautological* if it does not contain both a variable and its negation. The *size* of a clause is the number of literals in it. A *CNF formula*, or *CNF*, is a set of clauses, that we write as a conjunction of its elements. A k-CNF, where $k \geqslant 1$, is a CNF in which all clauses have size at most k. The *size* of a CNF F is the sum of the sizes of its clauses. We use $r(F)$ to denote the size of F.

An *assignment*, or *restriction*, is a partial map from the set of variables V to B. If α is an assignment and $X^{(b)}$ is a literal, then α *satisfies* $X^{(b)}$ if $X \in \mathrm{Dom}(\alpha)$ and $b = \alpha(X)$; it *falsifies* $X^{(b)}$ if $X \in \mathrm{Dom}(\alpha)$ and $b = 1 - \alpha(X)$. If C is a clause, then α *satisfies* C if it satisfies some literal of C; it *falsifies* C if it falsifies every literal of C. The *restriction of* C *by* α, denoted $C \restriction \alpha$, is 1 if α satisfies C and 0 if α falsifies C; if α neither satisfies nor falsifies C, then $C \restriction \alpha$ is the clause obtained from C by removing all the falsified literals of C, i.e., $C \restriction \alpha = C \setminus \{X^{(1-\alpha(X))} \mid X \in \mathrm{Dom}(\alpha)\}$. If F is a CNF, then $F \restriction \alpha$ is the CNF that contains $C \restriction \alpha$ for those $C \in F$ which are neither satisfied nor falsified by α, and that contains the empty clause if some $C \in F$ is falsified by α.

A clause D is a *weakening* of clause C if $C \subseteq D$. A clause E is a *resolvent* of clauses C and D if there is a variable X such that $X \in C$ and $\bar{X} \in D$, and $E = (C \setminus \{X\}) \cup (D \setminus \{\bar{X}\})$; we then speak of the resolvent of C and D *on* X, that we denote by $\mathrm{res}(C, D, X)$. We also say that E is obtained from C and D by a *cut on* X.

Let F be a CNF. A *Resolution proof from* F is a sequence $(D_1, \ldots, D_s)$ of non-tautological clauses, where $s \geqslant 1$ and, for all $u \in [s]$, it holds that D_u is a weakening of a clause in F, or there are $v, w \in [u-1]$ such that D_u is a weakening of a resolvent of D_v and D_w. The *length* of the proof is s; each D_u is a *line*. A *Resolution refutation of* F is a proof from F that ends with the empty clause, i.e., $D_s = \emptyset$. We let $s(F)$ denote the minimal s such that F has a Resolution refutation of length s; if F is satisfiable, we let $s(F) = \infty$. For a sequence of clauses $\Pi = (D_1, \ldots, D_s)$ let $\Pi \restriction \alpha$ be obtained from $(D_1 \restriction \alpha, \ldots, D_s \restriction \alpha)$ by removing 1's and replacing 0's by the empty clause. It is clear that if Π is a Resolution refutation of F of length s, then $\Pi \restriction \alpha$ is a Resolution refutation of $F \restriction \alpha$ of length at most s.

3 Refutations as structures

For this section we fix a CNF F with n variables $X_1, \ldots, X_n$ and m clauses $C_1, \ldots, C_m$. We view Resolution refutations $(D_1, \ldots, D_s)$ of F of length s as finite structures with a ternary relation D and four unary functions V, I, L, R:

$$\begin{aligned} &D \subseteq [s] \times [n] \times \mathrm{B}, \\ &V : [s] \to [n] \cup \{0\}, \\ &I : [s] \to [m] \cup \{0\}, \\ &L : [s] \to [s] \cup \{0\}, \\ &R : [s] \to [s] \cup \{0\}. \end{aligned} \tag{1}$$

The meaning of $(u, i, b) \in D$ is that the literal $X_i^{(b)}$ is in D_u. For each $u \in [s]$ exactly one of $V(u)$ or $I(u)$ is non-zero. The meaning of $V(u) = i \in [n]$ is that D_u is a weakening of the resolvent of D_v and D_w on X_i, where $v = L(u) \in [u-1]$ and $w = R(u) \in [u-1]$, and $\bar{X}_i \in D_v$ and $X_i \in D_w$. The meaning of $I(u) = j \in [m]$ is that D_u is a weakening of the clause C_j of F. Formally, a structure (D, V, I, L, R) of type (1) is a *refutation of* F *of length* s if the following hold for all $u, v \in [s]$, $i, i' \in [n]$, $j \in [m]$, and $b \in \mathrm{B}$:

(R1) $V(u) = 0$ or $I(u) = 0$, but not both;
(R2) if $I(u) = 0$, then both $R(u) \neq 0$ and $L(u) \neq 0$;
(R3) $L(u) < u$ and $R(u) < u$;
(R4a) if $V(u) = i$ and $L(u) = v$, then $(v, i, 0) \in D$;
(R4b) if $V(u) = i$ and $R(u) = v$, then $(v, i, 1) \in D$;
(R5a) if $V(u) = i \neq i'$, $L(u) = v$, and $(v, i', b) \in D$, then $(u, i', b) \in D$;
(R5b) if $V(u) = i \neq i'$, $R(u) = v$, and $(v, i', b) \in D$, then $(u, i', b) \in D$;
(R6) if $I(u) = j$ and $X_i^{(b)}$ appears in C_j, then $(u, i, b) \in D$;
(R7) $(u, i, 0) \notin D$ or $(u, i, 1) \notin D$;
(R8) $(s, i, b) \notin D$.

In words, (R1) determines, for every line D_u, whether it is a weakening of an initial clause, i.e., $I(u) \neq 0$, or a weakening of a resolvent, i.e., $V(u) \neq 0$. In the first case $C_{I(u)} \subseteq D_u$ by (R6). In the second case, $\mathrm{res}(D_{L(u)}, D_{R(u)}, X_{V(u)}) \subseteq D_u$ by (R4) and (R5), with (R2) and (R3) ensuring that $D_{L(u)}$ and $D_{R(u)}$ are earlier lines in the sequence. Finally, (R7) ensures no D_u is tautological, and (R8) ensures D_s is empty.

We give an example that will play a crucial role in the proof of the width lower bound.

Example 3. We use $(D^*, V^*, I^*, L^*, R^*)$ to denote the *full-tree Resolution refutation* of F. It has length

$$s^* := 2^{n+1} - 1$$

and its clauses are arranged in the form of a full binary tree of height n with $2^n - 1$ internal nodes and 2^n leaves. This tree has one node n_a at level $h \in \{0\} \cup [n]$ for every $a = (a_1, \dots, a_h) \in \{0, 1\}^h$ that is labelled by the clause

$$C_a = X_1^{(a_1)} \vee \cdots \vee X_h^{(a_h)},$$

that is, the unique clause in these variables falsified by the assignment that maps X_i to $1 - a_i$. In particular, the root of the tree is labelled by the empty clause and, for $h \in [n]$ and $a \in \{0, 1\}^{h-1}$, the clause C_a that labels node n_a is the resolvent of the clauses C_{a1} and C_{a0} that label the children nodes n_{a1} and n_{a0} on the variable X_h, i.e., $C_a = \mathrm{res}(C_{a1}, C_{a0}, X_h)$. Since F is unsatisfiable, every clause C_a that labels a leaf n_a is a weakening of some clause C_j of F.

To view this refutation as a structure of type (1) we have to identify the nodes n_a with numbers in $[s^*]$. We first identify the leafs, i.e., the nodes n_a with $a \in \{0, 1\}^n$, with the numbers $[2^n]$, then we identify the nodes on level $n - 1$, i.e., the nodes n_a with $a \in \{0, 1\}^{n-1}$, with the numbers in $[2^n + 2^{n-1}] \setminus [2^n]$ and so on, with the root getting $s^* = 2^{n+1} - 1$.

Let $a = (a_1, \dots, a_h) \in \{0, 1\}^h$ for $h \leqslant n$. We set $V^*(n_a) := 0$ if $h = n$, and $V^*(n_a) := h$ if $h < n$. We set $I^*(n_a) := 0$ if $h < n$, and $I^*(n_a) := j$ if $h = n$ and $j \in [m]$ is, say, smallest such that C_a is a weakening of C_j. We set $L^*(n_a) := R^*(n_a) := 0$ if $h = n$. If $h < n$ we set $L^*(n_a) := n_{a0}$ and $R^*(n_a) := n_{a1}$. Finally, $(n_a, i, b) \in D^*$ if and only if $i \in [h]$ and $b = a_i$.

4 Non-relativized formula REF

Given a CNF F with n variables $X_1, \dots, X_n$ and m non-tautological clauses $C_1, \dots, C_m$, and a natural number $s \geqslant 1$, we describe a CNF formula $\mathrm{REF}(F, s)$ that is satisfiable if and only if F has a refutation of length s. Its variables are:

- $D[u, i, b]$ for $u \in [s]$, $i \in [n]$, $b \in \mathrm{B}$ indicating that $(u, i, b) \in D$.
- $V[u, i]$ for $u \in [s]$, $i \in [n] \cup \{0\}$ indicating that $V(u) = i$.
- $I[u, j]$ for $u \in [s]$, $j \in [m] \cup \{0\}$ indicating that $I(u) = j$.
- $L[u, v]$ for $u \in [s]$, $v \in [s] \cup \{0\}$ indicating that $L(u) = v$.
- $R[u, v]$ for $u \in [s]$, $v \in [s] \cup \{0\}$ indicating that $R(u) = v$.

Clearly, any assignment to these variables describes a ternary relation D and binary relations V, I, L and R. The clauses of $\mathrm{REF}(F, s)$ are listed in Table 4 of Appendix A. This set of clauses is satisfied precisely by those assignments that describe refutations of F of length s. Conversely, given a structure as in (1) the *associated* assignment α satisfies $\mathrm{REF}(F, s)$ if and only if (D, V, I, L, R) is a refutation of F of length s; this assignment α maps variables $D[u, i, b]$, $V[u, i]$, $I[u, j]$, $L[u, v]$, and $R[u, v]$ to 1 or 0 depending on whether, respectively, $(u, i, b) \in D$, $V(u) = i$, $I(u) = j$, $L(u) = v$, and $R(u) = v$ or not.

The index $u \in [s]$ is *mentioned* in the variables

$$D[u, i, b], V[u, i], I[u, j], L[u, v], R[u, v].$$

Observe that if $v \neq u$, then v is not mentioned in $L[u, v]$ or $R[u, v]$. The *index-width* of a clause is the number of indices mentioned by some variable occurring in the clause. Observe that all clauses of $\mathrm{REF}(F, s)$ have index-width at most two. The index-width of a Resolution refutation is the maximum index-width of its clauses.

Lemma 4. *For all integers $n, w, s \geqslant 1$ with $2^n \geqslant s \geqslant 6nw$ and every unsatisfiable CNF F with n variables, every Resolution refutation of* $\mathrm{REF}(F, s)$ *has index-width at least w.*

Proof. Fix an unsatisfiable CNF F with n variables and m clauses. For this proof let G denote the formula $\mathrm{REF}(F, s)$ and let G^* denote the formula $\mathrm{REF}(F, s^*)$, where $s^* = 2^{n+1} - 1$ is the length of the full-tree Resolution refutation of F from Example 3, which exists for F because it is unsatisfiable. Let α^* be the assignment associated to $(D^*, V^*, I^*, L^*, R^*)$.

Let k be an integer such that $2^k < 3w \leqslant 2^{k+1}$ and note that $1 \leqslant k < n$ since $n, w \geqslant 1$ and $2^n \geqslant 6nw$. We partition $[s^*]$ into $n - k + 1$ intervals $B_0^*, B_1^*, \dots, B_{n-k}^*$ where

$$\begin{aligned} B_0^* &:= [s^*] \setminus [s^* - 2^{k+1} + 1], \\ B_i^* &:= [s^* - 2^{k+i} + 1] \setminus [s^* - 2^{k+1+i} + 1] \qquad \text{for } i = 1, \dots, n-k. \end{aligned}$$

In the notation of Example 3, $B_0^* = \{n_a \mid a \in \{0,1\}^{\leqslant k}\}$ is the set of $2^{k+1} - 1$ many nodes at the top k levels of the full binary tree. For $i \in [n-k]$, the i-th block $B_i^* = \{n_a \mid a \in \{0,1\}^{k+i}\}$ is the set of nodes at level $k+i$ of the full binary tree. In particular, B_{n-k}^* is the set of leaves.

Likewise, we partition $[s]$ into $n-k+1$ intervals $B_0, B_1, \ldots, B_{n-k}$ where

$$\begin{aligned}
&B_0 := [s] \setminus [s - 2^{k+1} + 1], \\
&B_i := [s - 2^{k+1} \cdot i + 1] \setminus [s - 2^{k+1} \cdot (i+1) + 1] \quad \text{for } i = 1, \ldots, n-k-1, \\
&B_{n-k} := [s - 2^{k+1} \cdot (n-k) + 1].
\end{aligned}$$

Observe that $|B_0^*| = |B_0| = 2^{k+1} - 1$; let $t : B_0 \to B_0^*$ be the bijection defined by $t(u) := u - s + s^*$ so that for all $u, v \in B_0$ it holds that

$$u < v \quad \text{if, and only if,} \quad t(u) < t(v). \tag{2}$$

Observe that for all $i \in [n-k-1]$:

$$|B_i^*| = 2^{k+i} \geqslant 2^{k+1} = |B_i| \geqslant 3w. \tag{3}$$

$$|B_{n-k}^*| = 2^n \geqslant s - 2^{k+1} \cdot (n-k) + 1 = |B_{n-k}| \geqslant 3w, \tag{4}$$

with the second following from $2^n \geqslant s \geqslant 6nw$ and $1 \leqslant k < n$.

Let $\mathscr{H}$ be the collection of partial functions $h : [s] \cup \{0\} \to [s^*] \cup \{0\}$ such that:

(H1) h is injective,
(H2) $0 \in \mathrm{Dom}(h)$ and $h(0) = 0$,
(H3) if $u \in \mathrm{Dom}(h) \cap B_0$, then $h(u) = t(u) \in B_0^*$,
(H4) if $u \in \mathrm{Dom}(h) \cap B_i$ with $i \in [n-k]$, then $h(u) \in B_i^*$.

In words, condition (H4) says that h preserves membership in matching intervals, and (H3) says that the 0-intervals are kept intact through the fixed bijection t. Preserving the intervals has the following important consequence:

Claim 5. *For every $h \in \mathscr{H}$ and $u, v \in \mathrm{Dom}(h) \setminus \{0\}$ the following hold:*

1. *$h(u) \neq 0$ and $h(v) \neq 0$,*
2. *if $L^*(h(v)) \in \mathrm{Img}(h)$, then $h^{-1}(L^*(h(v))) < v$,*
3. *if $R^*(h(v)) \in \mathrm{Img}(h)$, then $h^{-1}(R^*(h(v))) < v$.*

Proof. Property *1* follows from (H1) and (H2). To prove *2* we distinguish several cases: If $v \in B_{n-k}$, then $h(v) \in B_{n-k}^*$ by (H4), hence $L^*(h(v)) = 0$ and $h^{-1}(L^*(h(v))) = 0$ by (H2), which is smaller than $v \neq 0$. If $v \in B_i$ for some $i \in [n-k-1]$, then $h(v) \in B_i^*$ by (H4), hence $L^*(h(v)) \in B_{i+1}^*$, and $h^{-1}(L^*(h(v))) \in B_{i+1}$ by (H4) again, which is smaller than $v \in B_i$. If $v \in B_0$, then first note that $h(v) = t(v) \in B_0^*$ by (H3). We distinguish the cases whether $L^*(h(v)) \in B_0^*$ or not. In case $L^*(h(v)) \in B_0^*$, we have $h^{-1}(L^*(h(v))) = t^{-1}(L^*(h(v)))$. Since $L^*(h(v)) < h(v)$, by (2) we have $t^{-1}(L^*(h(v))) < t^{-1}(h(v)) = t^{-1}(t(v)) = v$. In case $L^*(h(v)) \notin B_0^*$, we have $L^*(h(v)) \in B_1^*$, so $h^{-1}(L^*(h(v))) \in B_1$ by (H4), which is smaller than $v \in B_0$. The proof of *3* is analogous to that of *2*. □

For a set $I \subseteq [s^*] \cup \{0\}$, let

$$\partial I := \big\{L^*(u) \mid u \in I \setminus \{0\}\big\} \cup \big\{R^*(u) \mid u \in I \setminus \{0\}\big\}.$$

A *condition* is a pair $p = (g, h)$, where g and h are functions in $\mathscr{H}$, such that

$$\begin{array}{ll} \text{(C1)} & g \subseteq h, \\ \text{(C2)} & \mathrm{Img}(h) = \mathrm{Img}(g) \cup \partial\mathrm{Img}(g). \end{array}$$

We say a condition $p' = (g', h')$ *extends* p if $h \subseteq h'$, i.e., h' extends h as a function. Observe, since $0 \in \mathrm{Dom}(g)$,

$$|\mathrm{Dom}(h)| \leqslant 3|\mathrm{Dom}(g)| - 2. \tag{5}$$

We define a partial truth assignment $\alpha(p)$ that sets the variables of G as follows. Note that if $D[u, i, b]$, $V[u, i]$, and $I[u, j]$ are variables of G, then $D[g(u), i, b]$, $V[g(u), i]$, and $I[g(u), j]$ are variables of G^* which are evaluated by α^*. The assignment $\alpha(p)$ is defined precisely on the variables of G that mention some $u \in \mathrm{Dom}(g)$. For such u it maps

– $D[u, i, b]$ to $\alpha^*(D[g(u), i, b])$, for all $i \in [n]$ and $b \in \mathrm{B}$;

– $V[u, i]$ to $\alpha^*(V[g(u), i])$, for all $i \in [n] \cup \{0\}$;

– $I[u, j]$ to $\alpha^*(I[g(u), j])$, for all $j \in [m] \cup \{0\}$;

– $L[u, v]$ to 1 or 0 indicating whether $v = h^{-1}(L^*(g(u)))$, for all $v \in [s] \cup \{0\}$;

– $R[u, v]$ to 1 or 0 indicating whether $v = h^{-1}(R^*(g(v)))$, for all $v \in [s] \cup \{0\}$.

Note that $L^*(g(u))$ and $R^*(g(u))$ belong to $\partial\mathrm{Img}(g) \subseteq \mathrm{Img}(h)$ for every $u \in \mathrm{Dom}(g)$, so h^{-1} is defined in the last two cases.

Clearly, if a condition p' extends p, then $\alpha(p) \subseteq \alpha(p')$. For $I \subseteq [s]$, the *restriction of* p *to* I, denoted $p\upharpoonright I$, is the pair (g^*, h^*) where g^* is the restriction of g to $I \cup \{0\}$, and h^* is the restriction of h with image $\mathrm{Img}(g^*) \cup \partial\mathrm{Img}(g^*)$.

Claim 6. *If p is a condition and $I \subseteq [s]$, then $p\upharpoonright I$ is a condition and $\alpha(p\upharpoonright I) \subseteq \alpha(p)$.*

Proof. The requirement that g' and h' belong to $\mathscr{H}$ is obviously satisfied since (H1)-(H4) are preserved by restrictions to subsets that contain 0. (C1) and (C2) are clear, so $p\upharpoonright I$ is a condition. The inclusion $\alpha(p\upharpoonright I) \subseteq \alpha(p)$ is clear since p extends $p\upharpoonright I$. □

Claim 7. *If $p = (g, h)$ is a condition with $|\mathrm{Dom}(g)| \leqslant w$ and $u \in [s]$, then there exists a condition $p' = (g', h')$ that extends p and such that* $\mathrm{Dom}(g') = \mathrm{Dom}(g) \cup \{u\}$.

Proof. We assume $u \notin \mathrm{Dom}(g)$ (otherwise we take $p' := p$) and set $g' := g \cup \{(u, u')\}$ for $u' \in [s^*]$ chosen as follows: if $u \in B_0$, take $u' := t(u)$; otherwise $u \in B_i$ for some $i \in [n-k]$ and we choose $u' \in B_i^* \setminus \mathrm{Img}(h)$. Note there exists u' as desired because $|B_i^*| \geqslant 3w$ by (3) or (4), so by (5)

$$|B_i^* \setminus \mathrm{Img}(h)| \geqslant |B_i^*| - 3 \cdot |\mathrm{Dom}(g)| + 2 > 0.$$

It is clear that $g' \in \mathscr{H}$. Write $v_0' := L^*(u')$ and $v_1' := R^*(u')$. We have to find $v_0, v_1 \in [s] \cup \{0\}$ such that $h' := h \cup \{(v_0, v_0'), (v_1, v_1')\} \in \mathscr{H}$. Assume at least one of v_0', v_1' is not in $\mathrm{Img}(h)$. Then it is distinct from 0 (i.e., $u' \notin B^*_{n-k}$), say it is in B^*_i. If $i = 0$, we find v_0, v_1 as the pre-images of v_0', v_1' under t. Otherwise $i \in [n-k]$ and we choose $v_0, v_1 \in B_i$ such that h' is injective. This can be done because $|B_i| \geqslant 3w$ by (3) or (4), so by (5)

$$|B_i \setminus \mathrm{Dom}(h)| \geqslant |B_i| - 3 \cdot |\mathrm{Dom}(g)| + 2 \geqslant 2.$$

It is clear that $h' \in \mathscr{H}$. □

Claim 8. *If p is a condition and C is a clause of G, then $C \upharpoonright \alpha(p) \neq 0$.*

Proof. Let $p = (g, h)$, write $\alpha := \alpha(p)$ and assume α is defined on all variables of C. Then g is defined on all indices mentioned by C. We distinguish by cases according to the type (A1)-(A21) of C.

- In case C is of type (A1), i.e., $C = \bigvee_{i \in [n] \cup \{0\}} V[u, i]$ for some $u \in \mathrm{Dom}(g)$, then $C \upharpoonright \alpha$ equals $(\bigvee_{i \in [n] \cup \{0\}} V[g(u), i]) \upharpoonright \alpha^*$ and this is 1 because $\bigvee_{i \in [n] \cup \{0\}} V[g(u), i]$ is a clause of G^*. Case (A2) is similar.
- In case (A3), ($u \in \mathrm{Dom}(g)$ and) α satisfies $L[u, v]$ for $v := h^{-1}(L^*(g(u))) \in [s] \cup \{0\}$. Note $L^*(g(u)) \in \partial\mathrm{Img}(g) \subseteq \mathrm{Img}(h)$, so v is well-defined. Hence $C \upharpoonright \alpha = 1$. Case (A4) is similar.
- In case (A5), $C \upharpoonright \alpha$ equals $(\bar{V}[g(u), i] \vee \bar{V}[g(u), i']) \upharpoonright \alpha^*$ and this is 1 because $\bar{V}[g(u), i] \vee \bar{V}[g(u), i']$ is a clause of G^*. Case (A6) is similar.
- In case (A7), v or v' is distinct from $h^{-1}(L^*(g(u)))$ and then, respectively, $L[u, v]$ or $L[u, v']$ is falsified by α. Hence $C \upharpoonright \alpha = 1$. Case (A8) is similar.
- In case (A9), $C \upharpoonright \alpha$ equals $(\bar{I}[g(u), 0] \vee \bar{V}[g(u), 0]) \upharpoonright \alpha^*$. But this is 1 since $\bar{I}[g(u), 0] \vee \bar{V}[g(u), 0]$ is a clause of G^*. Case (A10) is similar.
- In case (A11), note $\alpha(L[u, 0]) = 1$ implies $h^{-1}(L^*(g(u))) = 0$, so $L^*(g(u)) = 0$ by Claim 5 (1). Then $g(u)$ is a leaf and $I^*(g(u)) \neq 0$. Hence $0 = \alpha^*(I[g(u), 0]) = \alpha(I[u, 0])$, so $C \upharpoonright \alpha = 1$. Case (A12) is similar.
- In case (A13), note $\alpha(L[u, v]) = 1$ implies $v = h^{-1}(L^*(g(u)))$. But $h^{-1}(L^*(g(u))) = h^{-1}(L^*(h(u))) < u$ by (C1) and Claim 5 (2). Case (A14) is similar.
- In case (A15), $C \upharpoonright \alpha = 0$ implies $u, v \in \mathrm{Dom}(g)$ and $v = h^{-1}(L^*(g(u)))$. Hence $h(v) = g(v) = L^*(g(u))$ (by (C1)) and $\alpha^*(L[g(u), g(v)]) = 1$. Further, $C \upharpoonright \alpha = 0$ implies $\alpha^*(V[g(u), i]) = 1$ and $\alpha^*(D[g(v), i, 0]) = 0$. Hence α^* falsifies the clause $\bar{L}[g(u), g(v)] \vee \bar{V}[g(u), i] \vee D[g(v), i, 0]$ of G^*, a contradiction. Cases (A16)-(A18) are similar.
- In case (A19), $C \upharpoonright \alpha = 0$ implies that α^* falsifies the clause $\bar{I}[g(u), j] \vee D[g(u), i, b]$ of G^*, a contradiction. Case (A20) is similar.
- In case (A21), $\alpha(\bar{D}[s, i, b]) = 0$ implies $s \in \mathrm{Dom}(g)$ and α^* falsifies the $\bar{D}[g(s), i, b]$. But this is a clause of G^* since $g(s) = t(s) = s^*$ by (H3) – contradiction.

This finishes the proof of Claim 8. □

We are ready to finish the proof of the lemma. Let P be the set of conditions $p = (g, h)$ with $|\mathrm{Dom}(g)| \leqslant w$. Assume that there exists a Resolution refutation of $\mathrm{REF}(F, s)$ of index-width smaller than w. Let $p_0 = (g_0, h_0)$ where $g_0 = h_0 = \{(0, 0)\}$ and note that $\partial\mathrm{Img}(g_0) = \emptyset$, so $p_0 \in P$. The assignment $\alpha(p_0)$ is empty and falsifies the empty clause, the last clause of the refutation. Let E be the earliest clause in the refutation such that $E\restriction\alpha(p) = 0$ for some condition $p \in P$. In particular, $\alpha(p)$ is defined on all variables of E. By Claim 8, E is not a weakening of a clause from G. Hence, E is obtained by a cut of earlier clauses C and D on some variable. Let $u \in [s]$ be the index mentioned by this variable. Choose p' according to Claim 7. Then $\alpha(p')$ is defined on all variables in C, D, and E and extends the partial assignment $\alpha(p)$, so falsifies E. By soundness it falsifies C or D, say, it falsifies C. Let p'' be the restriction of p' to the indices mentioned in C. Then $\alpha(p'')$ falsifies C and $p'' \in P$ by Claim 6. This contradicts the choice of E. □

Remark 9. The width lower bound in the previous lemma does not have much to do with Resolution; a more general version can be formulated using the notions of semantic refutations and Poizat width from [5]. The notion of a Poizat tree is straightforwardly adapted to the many-sorted structures coding refutations. Define *index Poizat width* like Poizat width but using the *index height* of a Poizat tree: the maximum over its branches of the number of indices from $[s]$ appearing in queries of the branch. Then, the conclusion of the above lemma can be strengthened to: every semantic refutation of $\mathrm{REF}(F, s)$ contains a formula of index Poizat width at least $w/3$.

5 Relativized formula RREF

Given a CNF formula F with n variables and m clauses, and a natural number $s \geqslant 1$, we define the CNF formula $\mathrm{RREF}(F, s)$ as follows. We again write $X_1, \ldots, X_n$ for the variables and $C_1, \ldots, C_m$ for the clauses of F. The CNF formula $\mathrm{RREF}(F, s)$ has the same variables as $\mathrm{REF}(F, s)$ plus

- $P[u]$ for $u \in [s]$ indicating that u is an "active" index.

The clauses of $\mathrm{RREF}(F, s)$ are very similar to those of $\mathrm{REF}(F, s)$ with a few additional literals in each clause, and three additional types of clauses. For easiness of future reference, we explicitly listed the new set of clauses in Table 5 of Appendix A. In words, $\mathrm{RREF}(F, s)$ says that the lines indexed by its at most s active indices describe a Resolution refutation of F, and it does not put any restriction on the structure of the lines on inactive indices.

First we prove the lower bound:

Lemma 10. *There is an integer $n_0 \geqslant 0$ such that for all integers n and w with $n \geqslant n_0$ and $20 \leqslant w \leqslant 2^n/(13n)$ and every unsatisfiable CNF formula F with n variables, every Resolution refutation of* $\mathrm{RREF}(F, 13nw)$ *has length bigger than $2^{2w/5}$.*

Proof. Let F be an unsatisfiable CNF with n variables and m clauses and $20 \leqslant w \leqslant 2^n/(13n)$. Assume Π is a Resolution refutation of $\mathrm{RREF}(F,t)$ of length $\ell \leqslant 2^{2w/5}$ where $t := 13nw$. We derive a contradiction assuming at various places that n is large enough and this determines the constant n_0. It will be clear that it does not depend on F or w.

We define a random restriction ρ to (a subset of) the variables of $\mathrm{RREF}(F,t)$ by the following random experiment:

1. independently for every $u \in [t]$, map $P[u]$ to 1 or 0 each with probability $1/2$;
2. let A be the set of $u \in [t]$ for which $P[u]$ is mapped to 1;
3. for every $u \in A$ and $v \in [t] \setminus A$, map both $L[u,v]$ and $R[u,v]$ to 0;
4. independently for every $u \in [t] \setminus A$ and every variable that mentions u, map the variable to 1 or 0 each with probability $1/2$.

A literal that mentions $u \in [t]$ evaluates to 1 under ρ with probability at least $1/4$, namely in the event that $P[u]$ is mapped to 0 in step 1 and the right value is chosen in step 4. Thus, the probability that a clause of index-width at least w is not satisfied by ρ is at most $(3/4)^w$. By the union bound, the probability that $\Pi\upharpoonright\rho$ contains a clause of index-width at least w is at most $\ell \cdot (3/4)^w$, which is strictly less than $1/4$ for $\ell \leqslant 2^{2w/5}$ (here we use that $w \geqslant 20$). Note the clauses of $\Pi\upharpoonright\rho$ use variables of $\mathrm{REF}(F,t)$, so index-width is well-defined.

The cardinality of the random subset A is a symmetric binomial random variable with expectation $t/2 = 13nw/2$. By the Chernoff bound there is a real $\epsilon > 0$, independent of F and w, such that $|A| < 6nw$ with probability at most $2^{-\epsilon nw}$. For large enough n this is strictly less than $1/4$. Further, $P[t]$ is mapped to 1 with probability $1/2$. Thus, for large enough n, by the union bound, there exists a restriction ρ in the support of the above distribution, say, with associated set $A \subseteq [t]$, such that:

(i) $\Pi\upharpoonright\rho$ has index-width smaller than w;

(ii) $|A| \geqslant 6nw$;

(iii) ρ maps $P[t]$ to 1, so $t \in A$.

By (iii), ρ satisfies (A24). Also, $C\upharpoonright\rho = 1$ for C a clause of type (A22) or (A23) because this holds for every restriction in the support of the distribution.

Let $s = |A|$ and let $\mathrm{REF}(F,A)$ be defined as $\mathrm{REF}(F,s)$ except that we use A instead $[s]$ as index set, with t in the role of s. More precisely, $\mathrm{REF}(F,A)$ is obtained from $\mathrm{REF}(F,s)$ by a copy of variables: a variable is replaced by the variable (of $\mathrm{REF}(F,t)$) obtained by changing its index $u \in [s]$ (and $v \in [s]$) to the u-th (and the v-th) member of A.

We claim that for every clause $C \in \mathrm{RREF}(F,t)$ we either have $C\upharpoonright\rho = 1$ or $C\upharpoonright\rho \in \mathrm{REF}(F,A)$. We already checked this for (A22)-(A24) and are left with (A1)-(A21). For example, if C is a clause of type (A3), then $C\upharpoonright\rho = 1$ if $u \notin A$, and otherwise $C\upharpoonright\rho = L[u,0] \vee \bigvee_{v\in A} L[u,v]$ is a clause in $\mathrm{REF}(F,A)$. The case that C is of type (A4) is similar. The remaining cases are obvious. Thus, $\Pi\upharpoonright\rho$ is a Resolution refutation of $\mathrm{REF}(F,A)$ of length at most ℓ. By (i) and (ii), if n is large enough, this contradicts Lemma 4 (note $2^n \geqslant t \geqslant s$). □

The next lemma gives a polynomial upper bound on the length of Resolution refutations of RREF(F, s) when F is satisfiable. In fact, its second statement gives an upper bound that is possibly sublinear in the size of RREF(F, s). This second statement is not needed to prove Theorems 1 and 2.

Lemma 11. *There is a polynomial $p(s, n, m)$ such that for all integers $n, m, s \geqslant 1$ and every satisfiable CNF formula F with n variables and m clauses, there exists a Resolution refutation of* RREF(F, s) *of length at most $p(s, n, m)$. In fact, $p(s, n, m) \in O((snm)^2)$.*

Proof. Let F be a satisfiable CNF with variables $X_1, \dots, X_n$ and clauses $C_1, \dots, C_m$. Let $\alpha : \{X_1, \dots, X_n\} \to \mathrm{B}$ be an assignment that satisfies F. We derive the clauses

$$\mathit{True}(u) \ := \ \bar{P}[u] \vee D[u, 1, \alpha(X_1)] \vee \cdots \vee D[u, n, \alpha(X_n)]$$

for $u = 1, 2, 3, \dots, s$ in order. Then n many cuts with (A21) and one cut with (A24) yield the empty clause.

First, we derive, for all $u \in [s]$ and $j \in [m]$, as sm many weakenings of clauses of RREF(F, s), the auxiliary clauses

$$A_0(j, u) \ := \ \bar{I}[u, j] \vee \mathit{True}(u).$$

Since α satisfies F we can choose for every $j \in [m]$ some $i_j \in [n]$ such that $X_{i_j}^{(\alpha(X_{i_j}))}$ appears in C_j. Then $\bar{P}[u] \vee \bar{I}[u, j] \vee D[u, i_j, \alpha(X_{i_j})]$ is a clause of RREF(F, s), namely (A19). But $A_0(j, u)$ is a weakening of this.

We derive $\mathit{True}(u)$ for $u = 1$ through $s + m + 2$ many cuts. Through a sequence of s many cuts, starting at (A4) and using (A14) for all $v \in [s]$, get $\bar{P}[u] \vee R[u, 0]$. Cut this with (A12) to get $\bar{P}[u] \vee \bar{I}[u, 0]$. Cut this with (A2), followed by a sequence of m many cuts with all $A_0(j, u)$ for $j \in [m]$ to get $\mathit{True}(u)$.

Now assume $u > 1$ and $\mathit{True}(v)$ have been derived for all $v < u$. First, we derive for every $i \in [n]$ the auxiliary clause

$$A_1(i, u) := \begin{cases} \bar{V}[u, i] \vee L[u, 0] \vee \mathit{True}(u) & \text{if } \alpha(X_i) = 1 \\ \bar{V}[u, i] \vee R[u, 0] \vee \mathit{True}(u) & \text{if } \alpha(X_i) = 0. \end{cases}$$

We treat the case $\alpha(X_i) = 1$, the case $\alpha(X_i) = 0$ is analogous. Let $v \in [u - 1]$. Cut (A15) with $\bar{P}[v] \vee \bar{D}[v, i, 0] \vee \bar{D}[v, i, 1]$ of type (A20) on $D[v, i, 0]$ to get

$$\bar{P}[u] \vee \bar{P}[v] \vee \bar{L}[u, v] \vee \bar{V}[u, i] \vee \bar{D}[v, i, \alpha(X_i)].$$

Cut this with $\mathit{True}(v)$ on $D[v, i, \alpha(X_i)]$ to get

$$\bar{P}[u] \vee \bar{P}[v] \vee \bar{L}[u, v] \vee \bar{V}[u, i] \vee \textstyle\bigvee_{i' \in [n] \setminus \{i\}} D[v, i', \alpha(X_{i'})].$$

Cut this with (A17) on $D[v, i', \alpha(X_{i'})]$ for every $i' \in [n] \setminus \{i\}$, and then with with (A22) on $P[v]$ to get

$$\bar{P}[u] \vee \bar{L}[u, v] \vee \bar{V}[u, i] \vee \textstyle\bigvee_{i' \in [n] \setminus \{i\}} D[u, i', \alpha(X_{i'})]. \tag{6}$$

Now cut (A3) with this formula for all $v \in [u-1]$, and with (A13) for all $u \leqslant v \leqslant s$ to get the following subclause of $A_1(i,u)$

$$\bar{P}[u] \vee L[u,0] \vee \bar{V}[u,i] \vee \bigvee_{i' \in [n] \setminus \{i\}} D[u,i',\alpha(X_{i'})].$$

For every $v \in [u-1]$, the clause (6) is derived with $n+2$ cuts. Thus, $A_1(i,u)$ is derived with $(n+2)(u-1)+s$ many cuts. Doing this for all $i \in [n]$ amounts to $n(n+2)(u-1)+ns$ many cuts.

Having derived the auxiliary clauses $A_1(i,u)$ we now derive $\mathit{True}(u)$ in a sequence of $n+m+4$ cuts. In a sequence of n many cuts, cut (A1) with $A_1(i,u)$ for all $i \in [n]$, to get

$$V[u,0] \vee L[u,0] \vee R[u,0] \vee \mathit{True}(u).$$

Cut with (A9) on $V[u,0]$, then with (A11) on $L[u,0]$, and then with (A12) on $R[u,0]$ to get $\bar{I}[u,0] \vee \mathit{True}(u)$. Cut (A2) with this and then with $A_0(j,u)$ for all $j \in [m]$ in sequence to get $\mathit{True}(u)$ as desired.

In total, the refutation uses

$$(s+2+m)+(n+1)+\sum_{u=2}^{s}\big(n(n+2)(u-1)+ns+(n+m+4)\big)$$

many cuts: the first term counts the cuts in the derivation of $\mathit{True}(1)$, the second term counts the cuts to get the empty clause from $\mathit{True}(s)$, and each term in the big sum counts the cuts in the derivation of $\mathit{True}(u)$ for $u = 2, \ldots, s$. The length of the refutation is bounded by the number of cuts plus the sm weakenings to get the $A_0(j,u)$'s plus the number of clauses of RREF(F,s). But, in fact, the clauses (A7) and (A8) are not used by the given refutation, and REF(F,s) has at most $O((snm)^2)$ many other clauses. □

Remark 12. In the proof of the upper bound Lemma 11, the built-in linear order in the definition of RREF plays a crucial role. This refers to the side conditions $u \leqslant v$ in clauses (A13) and (A14) of the definition of RREF in Table 5 of Appendix A. Indeed, it is not hard see that if the linear order were not built-in but interpreted, through new propositional variables $O[u,v]$ and its corresponding clause axioms, then the resulting version of RREF(F,s) would be exponentially hard for resolution independently of the satisfiability or unsatisfiability of F. This follows from an "infinite model argument" similar to the proof of the main theorem in [19].

6 Proofs of the hardness results

In this section we derive Theorems 1 and 2 stated in the Introduction.

Proof of Theorem 2. It suffices to define G on 3-CNF formulas F with a sufficiently large number of variables n. Note $m \leqslant 8n^3$ for m the number of clauses of F. We set

$$G(F) := \text{RREF}(F, 13n^2).$$

Note $G(F)$ has size between $n^{1/q}$ and n^q for some constant $q > 0$. Thus, (a) follows from the first statement of Lemma 11 for some constant $c > 0$, and (b) follows from Lemma 10 for $w := n$ and some constant $d > 0$ (note that $20 \leqslant w \leqslant 2^n/(13n)$ for sufficiently large n). □

Our main result, Theorem 1, is implied by the more general statement below. We say that Resolution is *automatizable in time t* if there is an algorithm that, given an unsatisfiable CNF formula F, computes some Resolution refutation of F in time $t(r(F) + s(F))$. Recall that a function $t : \mathbb{N} \to \mathbb{N}$ is *time-constructible* if there is an algorithm that given 1^n (the string of n many 1's) computes $1^{t(n)}$ in time $O(t(n))$. We say that t is *subexponential* if $t(n) \leqslant 2^{n^{o(1)}}$.

Theorem 13. *Let $t : \mathbb{N} \to \mathbb{N}$ be time-constructible, non-decreasing and subexponential. If Resolution is automatizable in time t, then there are polynomials $q(n)$ and $r(n)$ and an algorithm that, given a 3-CNF formula F with n variables, decides in time $O(t(r(n)) + q(n))$ whether F is satisfiable.*

Proof. Assume that Resolution is automatizable in time t and choose c, d and G from Theorem 2. Let q be as in the proof given above, so $G(F)$ has size at most n^q for every 3-CNF formula F with a sufficiently large number n of variables.

Consider the following algorithm. Given a 3-CNF formula F with n variables, compute the formula $G(F)$ and run the automating algorithm for up to $t(n^q + n^{qc})$ steps. If the algorithm returns a Resolution refutation within the allotted time, then output 'satisfiable'. Else output 'unsatisfiable'.

It is clear that the algorithm runs in time $O(t(n^q + n^{qc}) + q(n))$ for some polynomial $q(n)$; here we use that t is time-constructible. It suffices to show that it is correct on 3-CNF formulas F with a sufficiently large number of variables n. If F is satisfiable, then by (a) of Theorem 2, $G(F)$ has a Resolution refutation of length at most n^{qc}, so the automating algorithm computes a refutation within the alloted time and we answer 'satisfiable'; here we use that t is non-decreasing. If F is unsatisfiable, then by (b) of Theorem 2, no Resolution refutation of $G(F)$ has length at most $O(t(n^q + n^{qc}))$, so the automating algorithm cannot compute one within the allotted time and we answer 'unsatisfiable'; here we use that t is subexponential. □

7 Concluding remarks

This final section contains the observation that, by a padding argument, the constants c and d in Theorem 2 can be chosen arbitrarily close to 1 and 2, respectively, and finishes with some questions.

Proof of Theorem 2 for $c = 1 + \epsilon$ and $d = 2 + \epsilon$. Given $\epsilon > 0$ we define $G(F, t)$ for a 3-CNF F and a natural $t > 0$ and verify (a) and (b) for $G(F) := G(F, t)$ assuming that t and n are sufficiently large; again, n denotes the number of variables of F. The meaning of "sufficiently large" for t will depend only on ϵ. Write $w := n^t$ and let $G(F, t) := \mathrm{RREF}'(F, 13nw)$ be obtained from $\mathrm{RREF}(F, 13nw)$ by deleting the clauses of type (A7) and (A8). As has been noted in the proof, Lemma 11 holds true for $\mathrm{RREF}'(F, 13nw)$ instead $\mathrm{RREF}(F, 13nw)$.

Since F has at most $8n^3$ clauses, the size $r(t)$ of $G(F,t)$ satisfies $n^{2t} \leqslant r(t) \leqslant n^{2t+c_0}$ for some constant $c_0 > 0$, a number independent of F and t. By Lemma 11, $s(G(F,t)) < n^{2t+c_1}$ for some constant $c_1 > 0$; but this is at most $r(t)^{1+\epsilon}$ if $t > c_1/2\epsilon$. By Lemma 10, $s(G(F,t)) > 2^{2n^t/5}$, but this is more than $2^{r(t)^{1/(2+\epsilon)}}$ if $t > c_0/\epsilon$. □

The reduction above falls short to rule out weak automatizability of Resolution. For this we would need that $s(G(F)) = \infty$ when F is unsatisfiable, i.e., that $G(F)$ is satisfiable, but this is unlikely to hold for a polynomial time G as it would put 3-SAT in co-NP. We refer to [4] for a proof of equivalence of the different characterizations of weak automatizability used here and in the Introduction. The main problem left open by the current work is to find more convincing evidence that Resolution is not weakly automatizable.

On the more technical side, we would like to know whether the formulas REF$(F, p(n))$ are hard for Resolution where F ranges over unsatisfiable CNF formulas with n variables and p is some fixed polynomial. We conjecture that this is the case but we only succeeded in establishing a width lower bound[3]. Of course, one can define analogous formulas P-REF(F, s) for any proof system P. For all we know it could be that such formulas P-REF$(F, p(n))$ are hard for strong proof systems P like Frege or Extended Frege. Of course, it would be a major breakthrough to prove this, even under some plausible computational hardness hypothesis. We refer to [31, Chapter 27] for a discussion.

A Formulas REF and RREF

In this appendix we include the detailed lists of clauses of the formulas REF and RREF. Recall, we use bars to denote the negation of the variables, e.g., $\bar{L}[u,v]$ denotes the negation of $L[u,v]$.

(A1)	$V[u,0] \vee V[u,1] \vee \cdots \vee V[u,n]$	$u \in [s]$,
(A2)	$I[u,0] \vee I[u,1] \vee \cdots \vee I[u,m]$	$u \in [s]$,
(A3)	$L[u,0] \vee L[u,1] \vee \cdots \vee L[u,s]$	$u \in [s]$,
(A4)	$R[u,0] \vee R[u,1] \vee \cdots \vee R[u,s]$	$u \in [s]$,
(A5)	$\bar{V}[u,i] \vee \bar{V}[u,i']$	$u \in [s]$, $i, i' \in [n] \cup \{0\}$, $i \neq i'$,
(A6)	$\bar{I}[u,j] \vee \bar{I}[u,j']$	$u \in [s]$, $j, j' \in [m] \cup \{0\}$, $j \neq j'$,
(A7)	$\bar{L}[u,v] \vee \bar{L}[u,v']$	$u \in [s]$, $v, v' \in [s] \cup \{0\}$, $v \neq v'$,
(A8)	$\bar{R}[u,v] \vee \bar{R}[u,v']$	$u \in [s]$, $v, v' \in [s] \cup \{0\}$, $v \neq v'$,
(A9)	$\bar{I}[u,0] \vee \bar{V}[u,0]$	$u \in [s]$,
(A10)	$I[u,0] \vee V[u,0]$	$u \in [s]$,
(A11)	$\bar{I}[u,0] \vee \bar{L}[u,0]$	$u \in [s]$,
(A12)	$\bar{I}[u,0] \vee \bar{R}[u,0]$	$u \in [s]$,
(A13)	$\bar{L}[u,v]$	$u, v \in [s]$, $u \leqslant v$,
(A14)	$\bar{R}[u,v]$	$u, v \in [s]$, $u \leqslant v$,
(A15)	$\bar{L}[u,v] \vee \bar{V}[u,i] \vee D[v,i,0]$	$u, v \in [s]$, $i \in [n]$,
(A16)	$\bar{R}[u,v] \vee \bar{V}[u,i] \vee D[v,i,1]$	$u, v \in [s]$, $i \in [n]$,

[3] Recently, M. Garlík confirmed the conjecture [22].

(A17)	$\bar{L}[u,v] \vee \bar{V}[u,i] \vee \bar{D}[v,i',b] \vee D[u,i',b]$	$u, v \in [s]$, $i, i' \in [n]$, $b \in \mathrm{B}$, $i' \neq i$,
(A18)	$\bar{R}[u,v] \vee \bar{V}[u,i] \vee \bar{D}[v,i',b] \vee D[u,i',b]$	$u, v \in [s]$, $i, i' \in [n]$, $b \in \mathrm{B}$, $i' \neq i$,
(A19)	$\bar{I}[u,j] \vee D[u,i,b]$	$u \in [s]$, $j \in [m]$, $X_i^{(b)} \in C_j$,
(A20)	$\bar{D}[u,i,0] \vee \bar{D}[u,i,1]$	$u \in [s]$, $i \in [n]$,
(A21)	$\bar{D}[s,i,b]$	$i \in [n]$, $b \in \mathrm{B}$.

Table 4: Clauses of REF.

In the clauses of REF, clauses (A1)-(A8) say V, I, L and R are functions with the appropriate domains and ranges, (A9)-(A10) express (R1), (A11)-(A12) express (R2), (A13)-(A14) express (R3), (A15)-(A16) express (R4), (A17)-(A18) express (R5), (A19) expresses (R6), (A20) expresses (R7), and (A21) expresses (R8).

(A1)	$\bar{P}[u] \vee V[u,0] \vee V[u,1] \vee \cdots \vee V[u,n]$	$u \in [s]$,
(A2)	$\bar{P}[u] \vee I[u,0] \vee I[u,1] \vee \cdots \vee I[u,m]$	$u \in [s]$,
(A3)	$\bar{P}[u] \vee L[u,0] \vee L[u,1] \vee \cdots \vee L[u,s]$	$u \in [s]$,
(A4)	$\bar{P}[u] \vee R[u,0] \vee R[u,1] \vee \cdots \vee R[u,s]$	$u \in [s]$,
(A5)	$\bar{P}[u] \vee \bar{V}[u,i] \vee \bar{V}[u,i']$	$u \in [s]$, $i, i' \in [n] \cup \{0\}$, $i \neq i'$,
(A6)	$\bar{P}[u] \vee \bar{I}[u,j] \vee \bar{I}[u,j']$	$u \in [s]$, $j, j' \in [m] \cup \{0\}$, $j \neq j'$,
(A7)	$\bar{P}[u] \vee \bar{L}[u,v] \vee \bar{L}[u,v']$	$u \in [s]$, $v, v' \in [s] \cup \{0\}$, $v \neq v'$,
(A8)	$\bar{P}[u] \vee \bar{R}[u,v] \vee \bar{R}[u,v']$	$u \in [s]$, $v, v' \in [s] \cup \{0\}$, $v \neq v'$,
(A9)	$\bar{P}[u] \vee \bar{I}[u,0] \vee \bar{V}[u,0]$	$u \in [s]$,
(A10)	$\bar{P}[u] \vee I[u,0] \vee V[u,0]$	$u \in [s]$,
(A11)	$\bar{P}[u] \vee \bar{I}[u,0] \vee \bar{L}[u,0]$	$u \in [s]$,
(A12)	$\bar{P}[u] \vee \bar{I}[u,0] \vee \bar{R}[u,0]$	$u \in [s]$,
(A13)	$\bar{P}[u] \vee \bar{L}[u,v]$	$u, v \in [s]$, $u \leqslant v$,
(A14)	$\bar{P}[u] \vee \bar{R}[u,v]$	$u, v \in [s]$, $u \leqslant v$,
(A15)	$\bar{P}[u] \vee \bar{P}[v] \vee \bar{L}[u,v] \vee \bar{V}[u,i] \vee D[v,i,0]$	$u, v \in [s]$, $i \in [n]$,
(A16)	$\bar{P}[u] \vee \bar{P}[v] \vee \bar{R}[u,v] \vee \bar{V}[u,i] \vee D[v,i,1]$	$u, v \in [s]$, $i \in [n]$,
(A17)	$\bar{P}[u] \vee \bar{P}[v] \vee \bar{L}[u,v] \vee \bar{V}[u,i] \vee \bar{D}[v,i',b] \vee D[u,i',b]$	$u, v \in [s]$, $i, i' \in [n]$, $b \in \mathrm{B}$, $i' \neq i$,
(A18)	$\bar{P}[u] \vee \bar{P}[v] \vee \bar{R}[u,v] \vee \bar{V}[u,i] \vee \bar{D}[v,i',b] \vee D[u,i',b]$	$u, v \in [s]$, $i, i' \in [n]$, $b \in \mathrm{B}$, $i' \neq i$,
(A19)	$\bar{P}[u] \vee \bar{I}[u,j] \vee D[u,i,b]$	$u \in [s]$, $j \in [m]$, $X_i^{(b)} \in C_j$,
(A20)	$\bar{P}[u] \vee \bar{D}[u,i,0] \vee \bar{D}[u,i,1]$	$u \in [s]$, $i \in [n]$,
(A21)	$\bar{P}[s] \vee \bar{D}[s,i,b]$	$i \in [n]$, $b \in \mathrm{B}$,
(A22)	$\bar{P}[u] \vee \bar{L}[u,v] \vee P[v]$	$u \in [s]$, $v \in [s]$,
(A23)	$\bar{P}[u] \vee \bar{R}[u,v] \vee P[v]$	$u \in [s]$, $v \in [s]$,
(A24)	$P[s]$.	

Table 5: Clauses of RREF

The clauses of RREF are the same as for REF but we add to each clause the literals $\bar{P}[u]$ with $u \in [s]$ mentioned by the clause. More precisely, $\bar{P}[u]$ is added to the clauses (A1)-(A14) and (A19) and (A20), both $\bar{P}[u]$ and $\bar{P}[v]$ are added to the clauses (A15)-(A18), and

$\bar{P}[s]$ is added to clause (A21). Further, we add three additional types of clauses numbered (A22)-(A24).

Acknowledgments

Both authors were partially funded by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement ERC-2014-CoG 648276 (AUTAR). First author partially funded by MICCIN grant TIN2016-76573-C2-1P (TASSAT3). We are grateful to Ilario Bonacina and Michal Garlik for their very useful comments on an earlier draft of this paper.

References

[1] M. Alekhnovich and A. A. Razborov. Resolution is not automatizable unless W[P] is tractable. SIAM Journal on Computing 38 (4): 1347-1363, 2008.

[2] M. Alekhnovich, S. R. Buss, S. Moran and T. Pitassi. Minimum propositional proof length is NP-hard to linearly approximate. The Journal of Symbolic Logic 66: 171-191, 2001.

[3] A. Atserias. The proof-search problem between bounded-width Resolution and bounded-degree semi-algebraic proofs. 16th International Conference on Theory and Applications of Satisfiability Testing (SAT'13), LNCS 7962, Springer, pp. 1-17, 2013.

[4] A. Atserias and M. L. Bonet. On the automatizability of Resolution and related propositional proof systems. Information and Computation 189 (2): 182-201, 2004.

[5] A. Atserias and E. Maneva. Mean-payoff games and propositional proofs. Information and Computation 209 (4): 664-691, 2011.

[6] A. Atserias and M. Müller. Partially definable forcing and bounded arithmetic. Archive for Mathematical Logic 54 (1): 1-33, 2015.

[7] A. Atserias, M. Lauria, and J. Nordström. Narrow proofs may be maximally long. ACM Transactions on Computational Logic 17 (3), 19:1-19:30, 2016.

[8] P. Beame and T. Pitassi. 1996. Simplified and improved resolution lower bounds. In 37th Annual Symposium on Foundations of Computer Science (FOCS'96), IEEE Computer Society, pp. 274-282. 1996.

[9] A. Beckmann, P. Pudlák and N. Thapen. Parity games and propositional proofs. ACM Transactions on Computational Logic 15 (2), article 17, 2014.

[10] E. Ben-Sasson and A. Wigderson. Short proofs are narrow – Resolution made simple. Journal of the ACM 48 (2): 149-169, 2001.

[11] M. L. Bonet, T. Pitassi and R. Raz. On interpolation and automatization for Frege systems. SIAM Journal on Computing 29 (6): 1939-1967, 2000.

[12] M. L. Bonet, C. Domingo, R. Gavaldà, A. Maciel and T. Pitassi. Non-automatizability of bounded-depth Frege proofs. Computational Complexity 13 (1-2): 47-68, 2004.

[13] S. R. Buss. On Gödel's theorems on lengths of proofs II: Lower bounds for recognizing k symbol provability. In P. Clote and J. Remmel (eds.), Feasible Mathematics II, Birkhäuser, pp. 57-90, 1995.

[14] C. S. Calude, S. Jain, B. Khoussainov, W. Li and F. Stephan. Deciding parity games in quasipolynomial time. 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC'17). ACM, pp. 252-263, 2017.

[15] Y. Chen and J. Flum. On the complexity of Gödel's proof predicate. The Journal of Symbolic Logic 75 (1): 239-254, 2010.

[16] Y. Chen, M. Grohe and M. Grüber. On parameterized approximability. 2nd International Workshop on Parameterized and Exact Computation (IWPEC'06), LNCS 4169, pp.109-120, 2006.

[17] Y. Chen and B. Lin. The constant inapproximability of the parameterized dominating set problem. SIAM Journal on Computing 48 (2): 513-533, 2019.

[18] S. A. Cook and R. A. Reckhow. The relative efficiency of propositional proof systems. Journal of Symbolic Logic 44 (1): 36-50, 1979.

[19] S. Dantchev and S. Riis. On relativisation and complexity gap for Resolution-based proof systems. Computer Science Logic (CSL'03), LNCS 2803, pp. 142-154, Springer, 2003.

[20] K. Eickmeyer, M. Grohe and M. Grüber. Approximation of natural W[P]-complete minimisation problems is hard. 23rd Annual IEEE Conference on Computational Complexity (CCC'08), College Park, MD, pp. 8-18, 2008.

[21] J. Flum and M. Grohe. Parameterized Complexity Theory. Springer, 2006.

[22] M. Garlík. Resolution lower bounds for refutation statements. 44th International Symposium on Mathematical Foundations of Computer Science (MFCS'19), LIPIcs 138, pp. 37:1-37:13, 2019.

[23] J. Grollmann and A. L. Selman. Complexity measures for public-key cryptosystems. SIAM Journal on Computing 17 (2): 309-335, 1988.

[24] L. Huang and T. Pitassi. Automatizability and simple stochastic games. In International Colloquium on Automata, Languages and Programming (ICALP'11), LNCS 6755, pp. 605-617, 2011.

[25] K. Iwama. Complexity of finding short resolution proofs. Mathematical Foundations of Computer Science (MFCS'97), LNCS 1295, pp. 309-318, 1997.

[26] I. Mertz, T. Pitassi and Y. Wei. Short proofs are hard to find. 46th International Colloquium on Automata, Languages and Programming (ICALP'19), LIPIcs 132, pp. 84:1-84:16, 2019.

[27] J. Krajíček. Lower bounds to the size of constant-depth propositional proofs. The Journal of Symbolic Logic 59 (1): 73-86, 1994.

[28] J. Krajíček. Bounded Arithmetic, Propositional Logic, and Complexity Theory. Encyclopedia of Mathematics and Its Applications 60, Cambridge University Press, 1995.

[29] J. Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. The Journal of Symbolic Logic 62 (2): 457-486, 1997.

[30] J. Krajíček. On the weak pigeonhole principle, Fundamenta Mathematicae 170 (1-3): 123-140, 2001.

[31] J. Krajíček. Forcing with random variables and proof complexity. London Mathematical Society Lecture Note Series 382, Cambridge University Press, 2011.

[32] J. Krajíček. Proof Complexity. Encyclopedia of Mathematics and Its Applications 170, Cambridge University Press, Cambridge - New York - Melbourne, 2019.

[33] J. Krajíček and P. Pudlák. Some consequences of cryptographical conjectures for S^1_2 and EF. Information and Computation 140 (1): 82-94, 1998.

[34] M. Müller and S. Szeider. The treewidth of proofs. Information and Computation 255 (1): 147-164, 2017.

[35] A. Urquhart. Von Neumann, Gödel and complexity theory. Bulletin of Symbolic Logic (16) 4: 516-530, 2010.

[36] P. Pudlák. On reducibility and symmetry of disjoint NP-pairs. Theoretical Computer Science 295: 323-339, 2003.

[37] A. A. Razborov. On provably disjoint NP-pairs. Basic Research in Computer Science (BRICS) Report Series 1 (36), 1994.