Resolution is Not Automatizable Unless W[P] is Tractable

Michael Alekhnovich*, Alexander A. Razborov†

October 2, 2001

Abstract

We show that neither Resolution nor tree-like Resolution is automatizable unless the class **W[P]** from the hierarchy of parameterized problems is fixed-parameter tractable by randomized algorithms with one-sided error.

1. Introduction

The analysis of potential usefulness of proof search heuristics and automated theorem proving procedures based on a proof system P amounts (on the theoretical level) to the following two basic questions:

Question 1. Which theorems in principle possess efficient P-proofs?

Question 2. How to find the optimal (or, at least, nearly the optimal) proof of a given theorem in P?

The traditional proof complexity mostly dealt, and still deals with the first question. Recently, however, there has been a growing interest in the second one, too. An additional motivation to study the complexity of finding optimal

*Institute for Advanced Study, Princeton, US, misha@ias.edu. Supported by AMIAS grant

†Institute for Advanced Study, Princeton, US and Steklov Mathematical Institute, Moscow, Russia, razborov@ias.edu. Supported by the State of New Jersey and NSF grant CCR-9987077

proofs comes from deep connections with efficient interpolation theorems; we refer the reader to the excellent survey [BP98] for more details about this and also as to a good starting point for learning more about propositional proof complexity in general.

One convenient framework for the theoretical study of Question 2 was proposed in [BPR00]. Namely, they called a proof system P *automatizable* if there exists a deterministic algorithm A which, given a tautology τ, returns its P-proof in time polynomial *in the size of the shortest P-proof of τ*. The definition of a quasi-automatizable proof system is given in the same way, but we only require the algorithm A to run in time which is quasi-polynomial (in the same parameter).

One advantage of this definition is that it allows us to completely disregard the first basic question on the *existence* of efficient P-proofs and indeed concentrate on *finding* efficient proofs *provided* they exist. In particular, the notion of automatizability makes perfect sense for those (weak) proof systems for which hard tautologies are already known. Moreover, the weaker is our system the more likely it seems to be automatizable. One possible explanation of this phenomenon comes from the connection between automatizability and efficient interpolation (every automatizable proof system has efficient interpolation, and the property of having efficient interpolation is indeed anti-monotone w.r.t. the strength of the system). Anyway, given this connection, the results from [KP98, BPR00] imply that Extended Frege and TC^0-Frege proof systems respectively are not automatizable assuming some widely believed cryptographic assumptions. [BDG$^+$99] extended the latter result to bounded-depth Frege but under a much stronger assumption.

We are primarily interested in the automatizability of Resolution and tree-like Resolution. It is worth noting that both systems possess efficient interpolation, therefore their non-automatizability can not be proved via techniques similar to [KP98, BPR00, BDG$^+$99]. Nonetheless, [Iwa97] proved that it is **NP**-hard to find the shortest resolution refutation. [ABMP98] proved that if $\mathbf{P} \neq \mathbf{NP}$ then the length of the shortest resolution refutation can not be approximated to within a constant factor (both for general and tree-like Resolution). Under the stronger assumption $\mathbf{NP} \not\subseteq \mathbf{QP}$ they were able to improve the ratio from an arbitrary constant to $2^{\log^{1-\epsilon} n}$ (later [DS99] obtained a better PCP characterization of **NP** that allows to prove the same bound for arbitrary $\epsilon \to 0$ modulo $\mathbf{P} \neq \mathbf{NP}$).

In the opposite direction, [BP96] observed that tree-like Resolution is quasi-automatizable. Thus, it is unlikely to show that this system is not automatizable modulo $\mathbf{P} \neq \mathbf{NP}$ conjecture, because it would imply quasi-polynomial algorithms for $\mathbf{NP}$ (in case of general Resolution this goal seems also tricky at the moment because there is only one known example [BG99] for which proof search algorithm of [BSW99] is working in more than quasi-polynomial time). Therefore, for any method of establishing non-automatizability which is applicable to tree-like Resolution, one needs to invoke some complexity framework in which the asymptotics $n^{O(1)}$ and $n^{\log n}$ are essentially different.

One natural example of such a framework is *parameterized complexity* by Downey and Fellows (see [DF98]) in which algorithms working in time $f(k)n^{O(1)}$ and n^k are considered different from the point of effectiveness (here k is an integer input parameter that is supposed to be an "arbitrarily large" constant). In this paper we prove that neither Resolution nor tree-like Resolution is automatizable unless the class $\mathbf{W[P]}$ lying very high in the hierarchy of parameterized problems is fixed-parameter tractable by a randomized algorithm with one-sided error (Theorem 4.3). Our proof goes by a reduction from the optimization problem MINIMUM MONOTONE CIRCUIT SATISFYING ASSIGNMENT (MMCSA for short) whose decision version is complete for the class $\mathbf{W[P]}$. An alternative hardness assumption is that there is no *deterministic* fixed-parameter algorithm which *approximates* MMCSA within any constant factor (Theorem 4.2). It is worth noting in this connection that we were able to relate to each other the hardness of finding *exact* and *approximate* solutions for MMCSA without using the PCP Theorem (see the proof of Theorem 4.3). This result can be interesting in its own.

The paper is organized as follows. Section 2 contains necessary preliminaries and definitions, in Section 3 we present our core reduction from MMCSA to automatizability of Resolution, and in Section 4 we use (sometimes non-trivial) self-improving techniques to prove our main results, Theorems 4.2 and 4.3. The paper is concluded with some open problems in Section 5.

2. Preliminaries

2.1. Resolution and automatizability

Let x be a Boolean variable, i.e. a variable that ranges over the set $\{0,1\}$. A *literal* of x is either x (denoted sometimes as x^1) or $\bar{x}$ (denoted sometimes as x^0). A *clause* is a disjunction of literals. A *CNF* is a conjunction of pairwise different clauses.

Let $f(x_1,\dots,x_n)$ be an arbitrary function (possibly, partial) from $\{0,1\}^n$ to some finite domain D. An *assignment to* f is a mapping $\alpha : \{x_1,\dots,x_n\} \to \{0,1\}$. A *restriction of* f is a mapping $\rho : \{x_1,\dots,x_n\} \to \{0,1,\star\}$. We denote by $|\rho|$ the number of assigned variables, $|\rho| \stackrel{\text{def}}{=} |\rho^{-1}(\{0,1\})|$. The *restriction of a function* f *or CNF* τ *by* ρ, denoted by $f|_\rho$ $[\tau|_\rho]$ is the function [CNF] obtained from f [τ, respectively] by setting the value of each $x \in \rho^{-1}(\{0,1\})$ to $\rho(x)$, and leaving each $x \in \rho^{-1}(\star)$ as a variable.

One of the simplest and the most widely studied propositional proof systems is *Resolution* which operates with clauses and has one rule of inference called *resolution rule*:

$$\frac{A \vee x \qquad B \vee \bar{x}}{A \vee B}.$$

A resolution proof is *tree-like* if its underlying graph is a tree. A *resolution refutation* of a CNF τ is a resolution proof of the empty clause from the clauses appearing in τ.

The *size* of a resolution proof is the overall number of clauses in it. For an unsatisfiable CNF τ, $S(\tau)$ $[S_T(\tau)]$ is the minimal size of its resolution refutation [tree-like resolution refutation, respectively]. Clearly, $S(\tau) \leq S_T(\tau)$.

The *width* $w(C)$ *of a clause* C is the number of literals in C. The *width* $w(\tau)$ *of a set of clauses* τ (in particular, the width of a resolution proof) is the maximal width of a clause appearing in this set. For a CNF τ, let $n(\tau)$ be the overall number of distinct variables appearing in it, and let $|\tau|$ be the overall number of occurrences of variables in τ, i.e., $|\tau| \stackrel{\text{def}}{=} \sum_{C\in\tau} w(C)$. For an unsatisfiable CNF τ, $w(\tau \vdash \emptyset)$ will denote the minimal width of its resolution refutation.

For n, a non-negative integer let $[n] \stackrel{\text{def}}{=} \{1,2,\dots,n\}$, and let $[n]^k \stackrel{\text{def}}{=} \{I \subseteq [n] \mid |I| = k\}$.

We will recall the general definition of automatizability from [BPR00] for the special cases of Resolution and tree-like Resolution.

Definition 2.1 Resolution [tree-like Resolution] is *(quasi-)automatizable* if there exists a deterministic algorithm A which, given an unsatisfiable CNF τ, returns its resolution refutation [tree like resolution refutation, respectively] in time which is (quasi-)polynomial in $|\tau| + S(\tau)$ [$|\tau| + S_T(\tau)$, respectively].

Remark 1 Note that we do not require that all clauses from τ must necessarily appear in its refutations, therefore, we can not a priori expect the inequality $S(\tau) \geq |\tau|$. This is why we must introduce the term $|\tau|$ into the bound on the running time of A when adapting the general definition of automatizability from [BPR00] to the case of Resolution.

2.2. Parameterized complexity and MMCSA problem

We refer the reader to [DF98, Fel01] for a good general introduction to the topic of parameterized complexity.

Definition 2.2 ([DF98, Definition 2.4]) The class **FPT** (that stands for *fixed parameter tractable*) of parameterized problems consists of all languages $L \subseteq \Sigma^* \times \mathbb{N}$ for which there exists an algorithm Φ, a constant c and a recursive function $f : \mathbb{N} \longrightarrow \mathbb{N}$ such that:

1. the running time of $\Phi(\langle x, k\rangle)$ is at most $f(k) \cdot |x|^c$;
2. $\langle x, k\rangle \in L$ iff $\Phi(\langle x, k\rangle) = 1$.

Thus an algorithm is considered to be feasible if it works in time polynomial in n and $f(k)$, where k is supposed to be much smaller than n, and f is an arbitrarily large (recursive) function. A similar feasibility issue arises in the theory of polynomial time approximation schemes (PTAS) for **NP**-hard problems: assume that we have an algorithm that approximates a given problem within arbitrary error $\epsilon > 0$ working in time $n^{O(1/\epsilon)}$. Is it possible to get rid of $1/\epsilon$ in the exponent and do it in time $f(1/\epsilon)n^{O(1)}$ (the algorithms which obey the latter bound on the running time are called EPTAS, efficient polynomial time approximation schemes)?

It turns out that this question is highly related to the fixed parameter tractability. Namely, the existence of EPTAS for a given problem implies an *exact* algorithm for the corresponding fixed parameter version (see [Baz95, CT97]).

To study the complexity of parameterized problems, special *parameterized* reductions (that preserve the property of being in **FPT**) are used. For any t the parameterized problem WEIGHTED t-NORMALIZED SATISFIABILITY is defined by restricting the classical SATISFIABILITY to a certain class of Boolean formulas, the parameter k bounding Hamming weights of the satisfying assignment we are searching for. The complexity class $\mathbf{W[t]}$ consists of all problems that can be reduced to WEIGHTED-t-NORMALIZED-SATISFIABILITY via parameterized reductions, and the class $\mathbf{W[P]}$ (where P stands for polynomial) includes all problems reducible to WEIGHTED CIRCUIT SATISFIABILITY. These definitions lead to the following *parameterized hierarchy*, in which every inclusion is believed to be strict:

$$\mathbf{FPT} \subseteq \mathbf{W[1]} \subseteq \mathbf{W[2]} ... \subseteq \mathbf{W[P]}.$$

In our paper we construct a randomized parameterized reduction from the automatizability of Resolution to the following optimization problem (MMCSA in what follows) that was introduced in [ABMP98].

Monotone Minimum Circuit Satisfying Assignment:
Instance: A monotone circuit C in n variables over the basis $\{\wedge, \vee\}$.
Solution: An assignment $a \in \{0,1\}^n$ such that $C(a) = 1$.
Objective function: $k(a)$ defined as the number of a_i's which are equal to 1.

By $k(C)$ we will denote the optimum value $k(a)$ of a solution a for an instance C of MMCSA.

The following easy observation was made in [ABMP98] ("self-improvement").

Proposition 2.3 *For every fixed integer $d > 0$ there exists a poly-time computable function π which maps monotone circuits into monotone circuits and such that $k(\pi(C)) = k(C)^d$ for all C.*

The decision version of MMCSA was considered in [DF98] (under the name WEIGHTED MONOTONE CIRCUIT SATISFIABILITY) in the context of parameterized complexity and was shown to be complete in the class $\mathbf{W[P]}$.

In order to formulate our main result, we need to introduce the obvious hybrid of the classes **R** and **FPT**:

Definition 2.4 The class **FPR** of parameterized problems consists of all languages $L \subseteq \Sigma^* \times \mathbb{N}$ for which there exists a probabilistic algorithm Φ, a constant c and a recursive function $f : \mathbb{N} \longrightarrow \mathbb{N}$ such that:

1. the running time of $\Phi(\langle x, k\rangle)$ is at most $f(k) \cdot |x|^c$;
2. if $\langle x, k\rangle \in L$ then $\mathbf{P}[\Phi(\langle x, k\rangle) = 1] \geq 1/2$;
3. if $\langle x, k\rangle \notin L$ then $\mathbf{P}[\Phi(\langle x, k\rangle) = 1] = 0$.

3. Main reduction from MMCSA to automatizability of Resolution

This section is entirely devoted to the proof of the following lemma.

Lemma 3.1 *There exists a poly-time computable function τ which maps any pair $\langle C, 1^m\rangle$, where C is a monotone circuit and m is an integer, to an unsatisfiable CNF $\tau(C, m)$ such that:*

$$S_T(\tau(C, m)) \leq |C| \cdot m^{O(\min\{k(C), \log m\})}$$

and

$$S(\tau(C, m)) \geq m^{\Omega(\min\{k(C), \log m\})}. \tag{1}$$

We begin the proof of Lemma 3.1 with describing CNFs that form the main building block in $\tau(C, m)$ and establishing their necessary properties. From now on fix a monotone circuit C in n variables $p_1, \ldots, p_n$. Let ℓ be another parameter, and let $\mathcal{A} \subseteq \{0, 1\}^\ell$. We will call vectors from $\mathcal{A}$ (usually represented as columns) *admissible* and call an 0-1 matrix with ℓ rows $\mathcal{A}$-*admissible* if all its columns are so. Consider the following combinatorial principle $\mathcal{P}_{C,\mathcal{A}}$:

$\mathcal{P}_{C,\mathcal{A}}$: *every $(\ell \times n)$ 0-1 $\mathcal{A}$-admissible matrix A contains a row $i \in [\ell]$ such that $C(a_{i1}, a_{i2}, \ldots, a_{in}) = 1$.*

Let us formulate one sufficient condition for $\mathcal{P}_{C,\mathcal{A}}$ to be true in the real world.

Definition 3.2 $d_1(\mathcal{A})$ is the maximal d such that for every d vectors from $\mathcal{A}$ there exists a position $i \in [\ell]$ in which all these vectors have 1.

$d_1(\mathcal{A})$ can be also easily characterized in terms of minimum covers. Namely, if we associate with every $\begin{pmatrix} a_1 \\ \vdots \\ a_\ell \end{pmatrix} \in \mathcal{A}$ the subset $\{i \in [\ell] \mid a_i = 0\}$ of $[\ell]$ then

$d_1(\mathcal{A})+1$ is exactly the minimal number of such sets needed to cover the whole $[\ell]$.

Lemma 3.3 *If $k(C) \leq d_1(\mathcal{A})$ then $\mathcal{P}_{C,\mathcal{A}}$ is true.*

Proof. Let A be an $(\ell \times n)$ 0-1 $\mathcal{A}$-admissible matrix. Let $a=(a_1,\ldots,a_n)$ be such that $C(a_1,\ldots,a_n)=1$ and $k(a)=k(C)$. Let $\mathcal{A}_0 \stackrel{\text{def}}{=} \left\{ \left(\begin{array}{c} a_{1j} \\ \vdots \\ a_{\ell j} \end{array} \right) \middle| \; a_j = 1 \right\}$. Since $|\mathcal{A}_0| \leq k(a) = k(C) \leq d_1(\mathcal{A})$, there exists $i \in [\ell]$ such that $a_{ij} = 1$ whenever $a_j = 1$. This means $a_{ij} \geq a_j$ for all $j \in [n]$ and implies $C(a_{i1},\ldots,a_{in})=1$ since A is monotone.■

The proof of Lemma 3.3 suggests that the optimal propositional proof of the principle $\mathcal{P}_{C,\mathcal{A}}$ should work by the exhaustive search through all $|\mathcal{A}|^{k(C)}$ possible placements of admissible vectors to the columns $\{j \mid a_j = 1\}$ and thus have size roughly $|\mathcal{A}|^{k(C)}$. Our task is to find an encoding of (the negation of) $\mathcal{P}_{C,\mathcal{A}}$ as a CNF so that we can prove tight upper and lower bounds on $S_T(\tau(C,\mathcal{A}))$ and $S(\tau(C,\mathcal{A}))$ of (roughly) this order. The encoding will also involve auxiliary surjective functions $F_1,\ldots,F_n : \{0,1\}^s \longrightarrow \mathcal{A}$, $f_1,\ldots,f_\ell : \{0,1\}^s \longrightarrow [r]$, where f_is are possibly partial, and s, r are some new parameters. They will be used for encoding vectors from $\mathcal{A}$ and elements from another finite set of r *controls* by binary strings of length s. These complications will be needed for the lower bounds purposes.

In order to not obstruct the proof with irrelevant details, we will define our CNFs $\tau(C,\mathcal{A},\vec{F},\vec{f})$ and establish their necessary properties in a situation which is more general than what will be actually needed for completing the proof of Lemma 3.1 (see page 13). If the reader prefers, he/she may think during the course of the proof that $\ell = m$ and $\mathcal{A}$ is an arbitrary set of vectors such that $d_1(\mathcal{A}) \geq \Omega(\log m)$ and (see Definition 3.6) $d_0(\mathcal{A}) \geq \Omega(\log m)$. Furthermore, $r = \log m$, $s = O(\log m)$ and F_j, f_i will be $(\log m)$-surjective in the sense of Definition 3.5.

Definition 3.4 Let $C(p_1,\ldots,p_n)$ be a monotone circuit, $\mathcal{A} \subseteq \{0,1\}^\ell$ be a set of vectors and $F_1,\ldots,F_n : \{0,1\}^s \longrightarrow \mathcal{A}$, $f_1,\ldots,f_\ell : \{0,1\}^s \longrightarrow [r]$ be surjective functions, where f_is are possibly partial. For every $j \in [n]$ and $\nu \in [s]$ we introduce a propositional variable x_j^ν, for every $i \in [\ell]$ and $\nu \in [s]$ introduce a variable y_i^ν, and for every $i \in [\ell]$, every "control" $c \in [r]$ and every vertex v of the circuit C introduce the variable z_{iv}^c.

For $j \in [n]$ and $\vec{a} \in \mathcal{A}$, let us denote by $[\mathrm{Column}_j = \vec{a}]$ the predicate $F_j(x_j^1, \ldots, x_j^s) = \vec{a}$. Likewise, for $i \in [\ell]$ and $c \in [r]$, let $[\mathrm{Control}_i = c]$ denote the predicate $f_i(y_i^1, \ldots, y_i^s) = c$.

The CNF $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ consists of all clauses that result from the expansion of the following Boolean predicates as CNFs:

$$(y_i^1, \ldots, y_i^s) \in \mathrm{dom}(f_i), \text{ for all } i \in [\ell]; \tag{2}$$

$$\left.\begin{array}{c} ([\mathrm{Column}_j = \vec{a}] \wedge [\mathrm{Control}_i = c]) \supset z_{i,p_j}^c \\ \text{for all } \vec{a} \in \mathcal{A},\ i \in [\ell] \text{ such that } a_i = 1 \text{ and all } j \in [n],\ c \in [r]; \end{array}\right\} \tag{3}$$

$$\left.\begin{array}{c} \Big([\mathrm{Control}_i = c] \wedge (z_{i,v'}^c * z_{i,v''}^c)\Big) \supset z_{iv}^c \\ \text{for all } i \in [\ell],\ c \in [r] \text{ and all internal nodes } v \\ \text{corresponding to the instruction } v \leftarrow v' * v'',\ * \in \{\wedge, \vee\}; \end{array}\right\} \tag{4}$$

$$[\mathrm{Control}_i = c] \supset \bar{z}_{i,v_{\mathrm{fin}}}, \text{ where } v_{\mathrm{fin}} \text{ is the output node of } C. \tag{5}$$

Informally, every assignment to x-variables encodes an $\mathcal{A}$-admissible matrix A and every assignment to y-variables satisfying (2) encodes controls $c_1, \ldots, c_\ell$ (one control per row). The axioms (3) and (4) inductively state that $z_{iv}^{c_i}$ is greater or equal than the result of computation of the node v on the ith row of the matrix A. Thus, $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ is unsatisfiable (for arbitrary surjective $\vec{F}, \vec{f}$) if and only if $P_{C,\mathcal{A}}$ is true.

Remark 2 If we are interested in general Resolution only, the construction of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ can be a bit simplified. Namely, we do not need the controls, and we can encode the value attained by a node v on the ith row in the same simple way as we did with $[\mathrm{Column}_j = \vec{a}]$. Then lower bounds from Lemma 3.7 will still go through, but the upper bound will become problematic for the tree-like case (although, it will still hold for general Resolution).

The only thing we need from $\vec{F}, \vec{f}$ is that their ontoness is preserved under restricting not too many variables.

Definition 3.5 We say that an onto (possibly partial) function $g : \{0,1\}^k \longrightarrow D$ is *r-surjective* if for any restriction ρ with $|\rho| \leq r$ the function $g|_\rho$ is still onto.

Finally, for the lower bound purposes we need a notion dual to $d_1(\mathcal{A})$.

Definition 3.6 $d_0(\mathcal{A})$ is the maximal d such that for every d positions $i_1, \ldots, i_d \in [\ell]$ there exists $\vec{a} \in \mathcal{A}$ such that $a_{i_1} = \ldots = a_{i_d} = 0$.

Now we are ready to formulate our main technical lemma.

Lemma 3.7 *Let C be a monotone circuit in n variables, $\mathcal{A} \subseteq \{0,1\}^\ell$, $|\mathcal{A}| = m$ and $F_1, \ldots, F_n : \{0,1\}^s \longrightarrow \mathcal{A}$, $f_1, \ldots, f_\ell : \{0,1\}^s \longrightarrow [r]$ be r-surjective functions, where f_is are possibly partial; ℓ, m, r, s arbitrary integer parameters. Then the following bounds hold.*

a) *If $k(C) \leq d_1(\mathcal{A})$ then $S_T(\tau(C, \mathcal{A}, \vec{F}, \vec{f})) \leq O\left(|C| \cdot 2^{s(k(C)+1)}\right)$.*

b) $w(\tau(C, \mathcal{A}, \vec{F}, \vec{f}) \vdash \emptyset) \geq \frac{r}{2} \cdot \min\{k(C), d_0(\mathcal{A})\}$.

c) $S(\tau(C, \mathcal{A}, \vec{F}, \vec{f})) \geq \exp\left(\Omega\left(\frac{r^2}{s} \cdot \min\{k(C), d_0(\mathcal{A})\}\right)\right)$.

Proof of Lemma 3.7. Part a). By formalizing the proof of Lemma 3.3. Let $k \stackrel{\text{def}}{=} k(C)$ and $a_1, \ldots, a_n$ be such that $C(a_1, \ldots, a_n) = 1$ and $k(a) = k$. Assume for simplicity that $a_1 = \ldots = a_k = 1$, $a_{k+1} = \ldots = a_n = 0$. Fix arbitrary admissible vectors $\vec{a}_1 \stackrel{\text{def}}{=} \begin{pmatrix} a_{i1} \\ \vdots \\ a_{\ell 1} \end{pmatrix}, \ldots, \vec{a}_k \stackrel{\text{def}}{=} \begin{pmatrix} a_{ik} \\ \vdots \\ a_{\ell k} \end{pmatrix}$ and, using the inequality $d_1(\mathcal{A}) \geq k$, pick up an arbitrary $i \in [\ell]$ (depending in general on $\vec{a}_1, \ldots, \vec{a}_k$) such that $a_{i1} = a_{i2} = \ldots = a_{ik} = 1$. We want to infer from $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ all clauses in the CNF expansion of

$$[\text{Column}_1 \neq \vec{a}_1] \vee \ldots \vee [\text{Column}_k \neq \vec{a}_k] \vee [\text{Control}_i \neq c] \qquad (6)$$

for all $c \in [r]$. It is fairly obvious how to do this efficiently in general Resolution. Namely, let V be the set of all nodes of the circuit C that are evaluated to 1 by the assignment $(1^k, 0^{n-k})$. Then we may proceed by induction on the construction of C and subsequently infer

$$([\text{Column}_1 = \vec{a}_1] \wedge \ldots \wedge [\text{Column}_k = \vec{a}_k] \wedge [\text{Control}_i = c]) \supset z_{iv}^c$$

for all $v \in V$ until we reach v_{fin}.

In order to get a *tree-like* proof, however, we should employ a dual (top-down) strategy. Namely, enumerate the set of vertices in some order which is consistent with the topology of C: $V = \langle v_1 = p_1, v_2 = p_2, \ldots, v_k = p_k, v_{k+1}, v_{k+2}, \ldots, v_t = v_{\mathrm{fin}} \rangle$; all wires between vertices in V go from the left to the right. Then, by a reverse induction on $\mu = t, t-1, \ldots, k$ we infer (all clauses in the CNF expansion of) $[\mathrm{Control}_i = c] \supset \left(\bar{z}^c_{1,v_1} \vee \ldots \vee \bar{z}^c_{\mu,v_\mu} \right)$. For $\mu = t$ this is (a weakening of) (5), and for the inductive step we resolve with the appropriate axiom in (4). Finally, when we descend to $[\mathrm{Control}_i = c] \supset \left(\bar{z}^c_{1,p_1} \vee \ldots \vee \bar{z}^c_{k,p_k} \right)$, we consecutively resolve with the corresponding axioms (3) to get rid of $\bar{z}^c_{j,p_j}$ and arrive at (6). Clearly, this resolution inference of every individual clause in (6) is tree-like and has size $O(|C|)$.

Finally, for every $i \in [\ell]$, every clause in the variables $\{ y_i^\nu \mid 1 \leq \nu \leq s \}$ appears in one of the CNFs resulting from the predicates $\{ [\mathrm{Control}_i \neq c] \mid c \in [r] \}$ or in (2), and every clause in the variables $\left\{ x_j^\nu \mid 1 \leq \nu \leq s \right\}$ appears in one of $[\mathrm{Column}_j \neq \vec{a}_j]$. This gives us an obvious tree-like refutation of the set of clauses (6) that has size $O(2^{s(k+1)})$. Combining this refutation with previously constructed inferences of (6) from $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$, we get the desired upper bound.

Part b). We follow the general strategy proposed in [BSW99]. Let Row_i be the set of axioms in $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ that correspond to this particular row i. For a clause C, let $\mu(C)$ be the smallest cardinality of $I \subseteq [\ell]$ such that $\{ \mathrm{Row}_i \mid i \in I \}$ (semantically) implies C. As in the previous proofs, $\mu(C)$ is subadditive, that is $\mu(C) \leq \mu(C_1) + \mu(C_2)$ whenever C is obtained from C_1, C_2 via a single application of the resolution rule. It is also obvious that $\mu(A) = 1$ for any axiom A in $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$.

We claim that $\mu(\emptyset) > d_0(\mathcal{A})$. Indeed, fix any $I \subseteq [\ell]$ with $|I| \leq d_0(\mathcal{A})$. We need to construct an assignment that satisfies all axioms in $\{ \mathrm{Row}_i \mid i \in I \}$. Pick $\vec{a}$ accordingly to Definition 3.6 in such a way that $\forall i \in I (a_i = 0)$. Assign every x_j^ν to α_j^ν, where $\alpha_j^1, \ldots, \alpha_j^s$ is an arbitrary vector such that $F_j(\alpha_j^1, \ldots, \alpha_j^s) = \vec{a}$; assign y_i^ν in an arbitrary way with the only requirement that they satisfy (2), and assign all z-variables to 0. This assignment will satisfy all axioms in $\{ \mathrm{Row}_i \mid i \in I \}$ which proves $\mu(\emptyset) > d_0(\mathcal{A})$.

Thus, any resolution refutation of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ must contain a clause C with $\frac{1}{2} d_0(\mathcal{A}) \leq \mu(C) \leq d_0(\mathcal{A})$, and we only need to show that this implies $w(C) \geq \frac{r}{2} \cdot \min\{k, d_0(\mathcal{A})\}$. Fix $I \subseteq [\ell]$ such that $\frac{1}{2} d_0(\mathcal{A}) \leq |I| \leq d_0(\mathcal{A})$, $\{ \mathrm{Row}_i \mid i \in I \}$ semantically implies C and I is minimal with this property.

If for every $i \in [I]$ the clause C contains at least r variables amongst $\{ y_i^\nu \mid \nu \in [s] \} \cup \{ z^c_{iv} \mid c \in [r],\ v \text{ is a node} \}$, we are done. Thus, suppose that this is not the case for some $i_0 \in I$. Fix an arbitrary assignment α that satisfies all axioms in $\{\mathrm{Row}_i \mid i \in I \setminus \{i_0\}\}$ and falsifies C (such an assignment exists due to the minimality of I).

Let J_0 consist of those $j \in [n]$ for which the clause C contains at least r variables from $\left\{x_j^\nu \mid \nu \in [s]\right\}$. If $|J_0| \geq k$, we are also done. If this is not the case, we will show how to alter the assignment α so that it will satisfy all axioms in $\{\mathrm{Row}_i \mid i \in I\}$ (including Row_{i_0}) but still falsify C, and this will give us the contradiction.

According to Definition 3.6, there exists $\vec{a} \in \mathcal{A}$ such that $a_i = 0$ for all $i \in I$. We alter α as follows.

1. For every $j \notin J_0$ we, using that F_j is r-surjective, change the values of the variables $\left\{x_j^\nu \mid \nu \in [s]\right\}$ not appearing in C in such a way that $F_j(x_j^1, \ldots, x_j^s) = \vec{a}$.

2. Pick any control $c_0 \in [r]$ such that no variable $z_{i0,v}^{c0}$ appears in C. Using the fact that f_{i_0} is r-surjective, change the values of variables $\left\{y_{i_0}^\nu \mid \nu \in [s]\right\}$ not appearing in C in such a way that $f_{i_0}(y_{i_0}^1, \ldots, y_{i_0}^s) = c_0$. Finally, re-assign every $z_{i0,v}^{c0}$ to the value computed by the node v on the characteristic vector of the set J_0. Note that z_{i0,p_j}^{c0} is set to 1 for $j \in J_0$ whereas $z_{i0,v_{\mathrm{fin}}}^{c0}$ is set to 0 since $|J_0| < k(C)$.

With the remarks made in this description, it is easy to see that the altered assignment will satisfy all axioms in $\{\mathrm{Row}_i \mid i \in I\}$. Also it will falsify C since we have not touched variables appearing in C. This contradiction with the fact that $\{\mathrm{Row}_i \mid i \in I\}$ implies C completes the proof of part b).

Part c). We apply the standard argument with width-reducing restrictions (cf. [BP96]). For doing this we observe that the CNFs of the form $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ well behave with respect to certain restrictions. Namely, let $d \leq r$ and $R \subseteq [r]$ be an arbitrary set of controls. Denote by $\mathcal{R}_{d,R}$ the set of all restrictions that arbitrarily assign to a Boolean value d variables in every one of the groups $\left\{x_j^\nu \mid \nu \in [s]\right\}, \{y_i^\nu \mid \nu \in [s]\}$ with $j \in [n], i \in [\ell]$ as well as all variables z_{iv}^c with $c \notin R$. Then it is easy to see that for $\rho \in \mathcal{R}_{d,R}$, every non-trivial clause in $\tau(C, \mathcal{A}, \vec{F}, \vec{f})|_\rho$, after a suitable re-enumeration of variables and controls, contains a subclause from $\tau(C, \mathcal{A}, \vec{F}|_\rho, (\vec{f}|_\rho)|_R)$ (the partial function $(f_i|_\rho)|_R$ is obtained from $f_i|_\rho$ by restricting its domain to $\{y_i \mid f_i|_\rho(y_i) \in R\}$ and range to R).

Pick now $\boldsymbol{\rho}$ uniformly and at random from $\mathcal{R}_{r/2,\boldsymbol{R}}$, where $\boldsymbol{R}$ is picked at random from $[r]^{r/2}$. Then $F_j|_{\boldsymbol{\rho}}, (f_i|_{\boldsymbol{\rho}})|_{\boldsymbol{R}}$ will be $(r/2)$-surjective. Therefore, by the already proven part b), for every refutation P of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$, $\boldsymbol{\rho}(P)$ will contain a clause of width $\Omega\left(r \cdot \min\{k(C), d_0(\mathcal{A})\}\right)$ with probability 1.

It is easy to see, however, that every clause of this width is killed by $\boldsymbol{\rho}$ with probability $\geq 1 - \exp\left(-\Omega\left(\frac{r^2}{s} \cdot \min\{k(C), d_0(\mathcal{A})\}\right)\right)$. Therefore, the size of P must be at least $\exp\left(\Omega\left(\frac{r^2}{s} \cdot \min\{k(C), d_0(\mathcal{A})\}\right)\right)$ since otherwise a random restriction $\boldsymbol{\rho}$ would have killed all such clauses with non-zero probability, which is impossible.

Lemma 3.7 is completely proved.$\blacksquare$

Proof of Lemma 3.1. Our construction of $\tau(C, m)$ proceeds in polynomial time as follows.

1. We may assume w.l.o.g. that m is a prime. Let P_m be the $(m \times m)$ 0-1 *Paley matrix* given by $a_{ij} = 1$ if and only if $j \neq i$ and $(j - i)$ is a quadratic residue mod m. Set $\ell \stackrel{\text{def}}{=} m$, and let $\mathcal{A} \subseteq \{0,1\}^m$ consist of all columns of P_m. Then $|\mathcal{A}| = m$ and $d_0(\mathcal{A}), d_1(\mathcal{A}) \geq \frac{1}{4} \log m$ (see e.g. [Alo95]).

2. Fix any poly-time computable $\mathbb{F}_2$-linear code $L \subseteq \{0,1\}^{h\lceil \log m \rceil}$ of dimension $\lceil \log m \rceil$ and with the minimal distance $\geq \lceil \log m \rceil$, where $h > 0$ is an absolute constant. Then the linear mapping $G : \{0,1\}^{h\lceil \log m \rceil} \longrightarrow \{0,1\}^{\lceil \log m \rceil}$ which is dual to the inclusion $L \to \{0,1\}^{h\lceil \log m \rceil}$ is $\lceil \log m \rceil$-surjective. Set $r \stackrel{\text{def}}{=} \lceil \log m \rceil$ and $s \stackrel{\text{def}}{=} h\lceil \log m \rceil$. Consider arbitrary (poly-time computable) surjective mappings $\Pi : \{0,1\}^{h\lceil \log m \rceil} \longrightarrow \mathcal{A}$, $\pi : \{0,1\}^{h\lceil \log m \rceil} \longrightarrow [r]$, and let $F_j \stackrel{\text{def}}{=} G \circ \Pi, f_i \stackrel{\text{def}}{=} G \circ \pi$ for all i, j.

3. Construct $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$. Note that the size of this CNF is polynomial in $|C|, \ell, 2^s$ which is polynomial in $|C|, m$ due to our choice of parameters.

At this point, Lemma 3.7 c) already implies (1) for $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$. The only remaining problem is that a priori we do not have the condition $k(C) \leq d_1(\mathcal{A})$ needed for part a) of Lemma 3.7. We circumvent this by a trick similar to the one used in [ABMP98].

Namely, let τ_m be a fixed unsatisfiable (poly-time constructible) CNF with $S(\tau_m), S_T(\tau_m) = m^{\theta(\log m)}$ (for example, one can take the Pigeonhole principle with $\sqrt{\log m}$ pigeons) and such that its set of variables is disjoint from the set of variables of $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$. We finally set $\tau(C, m) \stackrel{\text{def}}{=} \tau(C, \mathcal{A}, \vec{F}, \vec{f}) \wedge \tau_m$.

Since both $\tau(C, \mathcal{A}, \vec{F}, \vec{f})$ and τ_m satisfy the lower bound in (1), Efficient Interpolation Property for Resolution implies that $\tau(C, m)$ satisfies this bound, too.

Lastly, if $k(C) \leq \frac{1}{4}\log m$ then, since $d_1(\mathcal{A}) \geq \frac{1}{4}\log m$, we can apply Lemma 3.7 a) to get the required upper bound $S_T(\tau(C,m)) \leq |C| \cdot m^{O(k(C))}$. If, on the other hand, $k(C) \geq \frac{1}{4}\log m$, the necessary upper bound $S_T(\tau(C,m)) \leq m^{O(\log m)}$ simply follows from the upper bound for τ_m. This completes the proof of Lemma 3.1.■

4. Self-improvement and main results

In this section we combine Lemma 3.1 with some non-trivial self-improvement technique. This will show non-automatizability of Resolution and tree-like Resolution modulo plausible assumptions about the intractability of the parameterized hierarchy. First we need to get rid of the dummy parameter m in the statement of Lemma 3.1.

Lemma 4.1 *If either Resolution or tree-like Resolution is automatizable then there exists an absolute constant $h > 0$ and an algorithm Φ working on pairs $\langle C, k\rangle$, where C is a monotone circuit and k is an integer such that:*

1. *the running time of $\Phi(\langle C, k\rangle)$ is at most $\exp(O(k^2)) \cdot |C|^{O(1)}$;*
2. *if $k(C) \leq k$ then $\Phi(\langle C, k\rangle) = 1$;*
3. *if $k(C) \geq hk$ then $\Phi(\langle C, k\rangle) = 0$.*

Proof. Combining the reduction in Lemma 3.1 with an automatizing algorithm for either Resolution or tree-like Resolution, we get an integer-valued function $S(C,m)$ computable in time $\left(|C| \cdot m^{\min\{k(C), \log m\}}\right)^{h_0}$ and such that

$$m^{\epsilon\cdot\min\{k(C),\log m\}} \leq S(C,m) \leq \left(|C| \cdot m^{\min\{k(C),\log m\}}\right)^{h_1}$$

for some absolute constants $\epsilon, h_0, h_1 > 0$. Set the constant h in the statement in such a way that

$$h^2 > \frac{h_1}{\epsilon}(h+1). \tag{7}$$

Our algorithm Φ works as follows. We set

$$m \stackrel{\text{def}}{=} \begin{cases} 2^{hk} & \text{if } k \geq \sqrt{\log |C|}, \\ 2^{h(\log |C|/k)} & \text{otherwise.} \end{cases}$$

Note that in any case $\log m \geq hk$. Φ simulates $\left(|C| \cdot m^k\right)^{h_0}$ steps in the computation of $S(C,m)$, outputs 1 if the computation halts within this time and its result $S(C,m)$ satisfies the inequality $S(C,m) \leq \left(|C| \cdot m^k\right)^{h_1}$, and outputs 0 in all other cases.

Our choice of m ensures that $m^k \leq \max\{2^{k^2}, |C|\}^h$, which implies property 1).

The upper bounds we have imposed on the running time and the output value of the algorithm Φ are the same as the bounds known for the underlying algorithm computing $S(C,m)$, the only difference is that we have replaced $k(C)$ by k. This observation implies property 2).

Finally, since $\log m \geq hk$, $k(C) \geq hk$ implies that $S(C,m) \geq m^{\epsilon kh}$, and elementary calculations show that, along with (7), this gives us $S(C,m) > \left(|C| \cdot m^k\right)^{h_1}$. Thus, if $k(C) \geq hk$, the algorithm Φ outputs the value 0.

Lemma 4.1 is proved.$\blacksquare$

Combining this lemma with Proposition 2.3, we immediately get our first main result.

Theorem 4.2 *If either Resolution or tree-like Resolution is automatizable then for any fixed $\epsilon > 0$ there exists an algorithm Φ working on monotone circuits C which runs in time $\exp\left(k(C)^{O(1)}\right) \cdot |C|^{O(1)}$ and approximates the value of $k(C)$ to within a factor $(1+\epsilon)$.*

Proof. Firstly we extract from Lemma 4.1 an algorithm which meets the bound on the running time and achieves the ratio of approximation h. For doing that we consecutively run the algorithm Φ from that lemma on the inputs $\langle C, 1\rangle, \ldots, \langle C, k\rangle, \ldots$, and output the first value k for which we get the answer 0.

Combining this algorithm with the self-improving reduction from Proposition 2.3 (for $d = \lceil \frac{1}{\epsilon} \ln h \rceil$), we get approximating algorithm with the required properties.$\blacksquare$

In the established terminology, what we have seen so far under the assumption of automatizability of (tree-like) Resolution is a PTAS (polynomial time approximation scheme) for MMCSA in the context of parameterized complexity (the latter referring to the term $\exp\left(k(C)^{O(1)}\right)$ in the bound on the running time). As we noted in Section 2.2, in this context *efficient* polynomial time approximation schemes lead to *exact* algorithms. The task of converting an arbitrary PTAS into an EPTAS seems to be hopeless in general even in the context of parameterized complexity. We nonetheless can perform it for the specific problem MMCSA using a much trickier self-improvement construction. This construction which appears in the proof of the next theorem might be of independent interest.

Recall Definition 2.4 of the class **FPR**.

Theorem 4.3 *If either Resolution or tree-like Resolution is automatizable then* $\mathbf{W}[\mathbf{P}] \subseteq co - \mathbf{FPR}$.

Proof. Let C be a monotone circuit in n variables and k be an integer such that

$$10 \leq k \leq \epsilon(\log n / \log\log n)^2 \tag{8}$$

for a sufficiently small constant $\epsilon > 0$ (we will remark later how to get rid of this condition). Our goal is to construct in polynomial time a randomized monotone circuit $\boldsymbol{\pi}(C, k)$ and an integer $\alpha(k)$ (deterministically depending only on k) such that α is recursive and the following conditions hold:

$$k(C) \leq k \implies \mathbf{P}[k(\boldsymbol{\pi}(C,k)) \leq \alpha(k)] = 1; \tag{9}$$

$$k(C) \geq k+1 \implies \mathbf{P}[k(\boldsymbol{\pi}(C,k)) \geq 2\alpha(k)] \geq 1/2. \tag{10}$$

Firstly we apply to C the reduction from Proposition 2.3 with $d = 2$. Re-denoting k^2 back to k, we may assume w.l.o.g. that in (10) we have a stronger premise:

$$k(C) \geq k + 2\sqrt{k} \implies \mathbf{P}[k(\boldsymbol{\pi}(C,k)) \geq 2\alpha(k)] \geq 1/2. \tag{11}$$

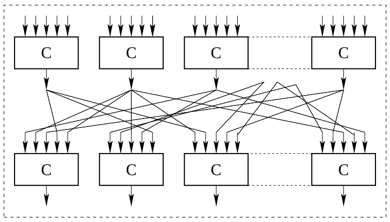

Figure 1: One layer of $\boldsymbol{\pi}(C, N, d)$

Now comes our main reduction. Let N, d be two parameters (to be specified later). The randomized circuit $\boldsymbol{\pi}(C, N, d)$ in (nN) variables consists of

d layers. Each layer consists of N independent copies of the circuit C (see Fig. 1); thus, it has (nN) inputs and N outputs. We connect input nodes at the $(i+1)$st level to output nodes at the ith level at random. Finally, we pick up an arbitrary output node at the last dth level and declare it to be the output of the whole circuit $\boldsymbol{\pi}(C, N, d)$.

Clearly, this construction is polynomial in $|C|, N, d$. Also, an obvious induction on d shows that $k(\boldsymbol{\pi}(C, N, d)) \leq k(C)^d$ with probability 1. In order to get a lower bound on $k(\boldsymbol{\pi}(C, N, d))$, we need the following easy lemma.

Lemma 4.4 *Let* $\boldsymbol{\chi} : [N] \times [n] \longrightarrow [N]$ *be a randomly chosen function, and* k, a *be any parameters. Then* $\mathbf{P}\Big[\exists V \in [N]^k (|\boldsymbol{\chi}(V \times [n])| \leq kn - a)\Big] \leq N^k \cdot \left(\frac{4k^2n^2}{N}\right)^a$.

Proof of Lemma 4.4. This event takes place if and only if there exist $V \in [N]^k$ and disjoint $D_1, \ldots, D_r \subseteq V \times [n]$ such that $|D_1|, \ldots, |D_r| \geq 2$, $\sum_{i=1}^r (|D_i| - 1) = a$ and $\boldsymbol{\chi}|_{D_i} = \text{const}$ for all $i \in [r]$. Since the two first properties imply $\sum_{i=1}^r |D_i| \leq 2a$, the overall number of all choices of $\langle D_1, \ldots, D_r \rangle$ does not exceed $N^k \cdot (2kn)^{2a}$. On the other hand, for every fixed choice we have

$$\mathbf{P}[\boldsymbol{\chi}|_{D_1} = \text{const}, \ldots, \boldsymbol{\chi}|_{D_r} = \text{const}] = \frac{N^r}{N^{(\sum_{i=1}^r |D_i|)}} = N^{-a}.$$

Lemma 4.4 follows.■

Now we can complete the description of our reduction. Namely, we set $N \stackrel{\text{def}}{=} n^3$, $d \stackrel{\text{def}}{=} \sqrt{k}$ and let $\boldsymbol{\pi}(C, k) \stackrel{\text{def}}{=} \boldsymbol{\pi}(C, n^3, \sqrt{k})$, $\alpha(k) \stackrel{\text{def}}{=} k^{\sqrt{k}}$.

(9) follows from the remark made above.

In order to check (11), denote by $\boldsymbol{\chi_i} : [N] \times [n] \longrightarrow [N]$ the function used for connecting input nodes at the $(i+1)$st level of $\boldsymbol{\pi}(C, n^3, \sqrt{k})$ to the output nodes at the ith level. Let $k_i \stackrel{\text{def}}{=} (k + \sqrt{k})^{d-i}$. Let us call $\boldsymbol{\pi}(C, N, d)$ *bad* if for at least one of these functions $\boldsymbol{\chi_i}$ there exists a set V of circuits at the $(i+1)$st level such that $|V| = k_{i+1}$ and $|\boldsymbol{\chi_i}(V \times [n])| \leq k_{i+1}(n - \sqrt{k})$. Using Lemma 4.4 and (8), we get the bound

$$\begin{aligned}
\mathbf{P}[\boldsymbol{\pi}(C, N, d) \text{ is bad }] &\leq \sum_{i=1}^{d-1} N^{k_{i+1}} \cdot \left(\frac{4k_{i+1}^2 n^2}{N}\right)^{\sqrt{k} \cdot k_{i+1}} \\
&= \sum_{i=1}^{d-1} \left(\frac{4k_{i+1}^2}{n^{1-3/\sqrt{k}}}\right)^{\sqrt{k} \cdot k_{i+1}} \leq 1/2.
\end{aligned}$$

On the other hand, it is easy to see by induction on $i = d, \ldots, 1$ that if $k(C) \geq k + 2\sqrt{k}$ and $\boldsymbol{\pi}(C, N, d)$ is good then every satisfying assignment a should satisfy at least k_i output nodes at the ith level. Indeed, the base $i = d$ is obvious. For the inductive step, assume that a satisfies the output nodes of a set V of circuits at the $(i+1)$th level, $|V| = k_{i+1}$. Then at least $(k + 2\sqrt{k}) \cdot k_{i+1}$ *input* nodes to these circuits should be satisfied. Since $\boldsymbol{\chi_i}$ is good, there are at most $\sqrt{k} \cdot k_{i+1}$ collisions between the $(k + 2\sqrt{k}) \cdot k_{i+1}$ wires leading to these nodes from the ith level. Therefore, at least $(k + 2\sqrt{k}) \cdot k_{i+1} - \sqrt{k} \cdot k_{i+1} = k_i$ output nodes at the ith level should be satisfied.

In particular, at the first level we will have $\geq (k+\sqrt{k})^{d-1}$ satisfied circuits and $\geq (k+2\sqrt{k}) \cdot (k+\sqrt{k})^{\sqrt{k}-1} > 2\alpha(k)$ satisfied input nodes. This completes the proof that our probabilistic reduction $\boldsymbol{\pi}(C, k)$ has the properties (9), (11).

Now we finish the proof of Theorem 4.3. Suppose that either Resolution or tree-like Resolution is automatizable. Since WEIGHTED MONOTONE CIRCUIT SATISFIABILITY is $\mathbf{W[P]}$-complete (see [DF98, Chapter 13]), we only have to show that the language $\{\langle C, k\rangle \mid k(C) \leq k\}$ is in $co-\mathbf{FPR}$. Given an input $\langle C, k\rangle$ we check the condition (8). If it is violated, we apply the straightforward brute-force algorithm with running time $O(|C| \cdot n^k) \leq |C| \cdot f(k) \cdot n^9$. Otherwise we simply combine our probabilistic reduction $\langle \boldsymbol{\pi}, \alpha\rangle$ with the deterministic algorithm for deciding whether $k(\boldsymbol{\pi}(C, k)) \leq \alpha(k)$ or $k(\boldsymbol{\pi}(C, k)) \geq 2\alpha(k)$ provided by Theorem 4.2. Theorem 4.3 is completely proved.$\blacksquare$

5. Open Problems

The main problem left open by this paper is whether general Resolution is quasi-automatizable. Since the width algorithm by Ben-Sasson and Wigderson [BSW99] finds a resolution refutation of any unsatisfiable CNF τ in time $n^{O(w(\tau \vdash \emptyset))}$, a negative solution to this problem must involve a construction of a broad and "tractable" family of CNF τ for which $S(\tau)$ is much smaller than $2^{w(\tau \vdash \emptyset)}$. Such families are not so easy to come by (e.g. our techniques involve showing the *opposite* in the proof of Lemma 3.7 c)), and it seems the only tautologies of this kind we have at the moment come from the somewhat isolated example [BG99].

We were not able to de-randomize the proof of Lemma 4.4. In the terminology of [ABSRW00], we need explicit constructions of $(N \times N)$ 0-1 matrices which would be $(k, n, n - O(1))$-expanders for $n \geq N^{\Omega(1)}$ and an arbitrary function $k = k(N)$ tending to infinity. Explicit constructions based on Ramanujan graphs seem to give only $(k, n, n - k^{\epsilon})$-expanders for any *fixed* ϵ which is not sufficient for our purposes. Can we weaken the hardness assumption in Theorem 4.3 to $\mathbf{W[P]} \neq \mathbf{FPT}$ by an explicit construction of better expanders (or by using any other means)?

References

[ABMP98] M. Alekhnovich, S. Buss, S. Moran, and T. Pitassi. Minimum propositional proof length is **NP**-hard to linearly approximate. In Lubos Brim, Jozef Gruska, and Jirí Zlatuska, editors, *Proceedings of the 23rd International Symposium on the Mathematical Foundations of Computer Science (Brno, August '98), Lecture Notes in Computer Science* 1450, pages 176–184. Springer-Verlag, 1998.

[ABSRW00] M. Alekhnovich, E. Ben-Sasson, A. Razborov, and A. Wigderson. Pseudorandom generators in propositional complexity. In *Proceedings of the 41st IEEE FOCS*, 2000.

[Alo95] N. Alon. Tools from higher algebra. In *Handbook of Combinatorics, vol. II*, chapter 32. Elsevier, 1995.

[Baz95] C. Bazgan. Schémas d'approximation et complexité paramétrée. Rapport de stage de DEA d'Informatique à Orsay, 1995.

[BDG+99] M. Bonet, C. Domingo, R. Gavaldá, A. Maciel, and T. Pitassi. Non-automatizability of bounded-depth Frege proofs. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, pages 15–23, 1999.

[BG99] M. Bonet and N. Galesi. A study of proof search algorithms for Resolution and Polynomial Calculus. In *Proceedings of the 40th IEEE FOCS*, pages 422–431, 1999.

[BP96] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *Proceedings of the 37th IEEE FOCS*, pages 274–282, 1996.

[BP98] P. Beame and T. Pitassi. Propositional proof complexity: Past, present and future. Technical Report TR98-067, Electronic Colloquium on Computational Complexity, 1998.

[BPR00] M. Bonet, T. Pitassi, and R. Raz. On interpolation and automatization for Frege systems. *SIAM Journal on Computing*, 29(6):1939–1967, 2000.

[BSW99] E. Ben-Sasson and A. Wigderson. Short proofs are narrow - resolution made simple. In *Proceedings of the* 31*st ACM STOC*, pages 517–526, 1999.

[CT97] M. Cesati and L. Trevisan. On the efficiency of polynomial time approximation schemes. *Information Processing Letters*, 64(4): 165–171, 1997.

[DS99] I. Dinur and S. Safra. On the hardness of approximating label cover. *Electronic Colloquium on Computational Complexity*, Technical report TR99-015, 1999.

[DF98] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1998.

[Iwa97] K. Iwama. Complexity of finding short resolution proofs. In P. Ružička I.Prívara, editor, *Proceedings of the* 22*nd International Symposium on the Mathematical Foundations of Computer Science* (*Bratislava, August* '97)*, Lecture Notes in Computer Science* 1295, pages 309–318. Springer-Verlag, 1997.

[Fel01] M. Fellows. Parameterized complexity: new developments and research frontiers. To appear in *Aspects of Complexity*, R. Downey and E. Hirshfeldt, editors, De Gruyter, 2001.

[KP98] J. Krajíček and P. Pudlák. Some consequences of cryptographical conjectures for S_2^1 and EF. *Information and Computation*, 142:82–94, 1998.