The Journal of Symbolic Logic
Volume xx, Number x, xxxx xxxx

MINIMUM PROPOSITIONAL PROOF LENGTH IS NP-HARD TO LINEARLY APPROXIMATE

MICHAEL ALEKHNOVICH, SAM BUSS, SHLOMO MORAN, AND TONIANN PITASSI

Abstract. We prove that the problem of determining the minimum propositional proof length is NP-hard to approximate within a factor of $2^{\log^{1-o(1)} n}$. These results are very robust in that they hold for almost all natural proof systems, including: Frege systems, extended Frege systems, resolution, Horn resolution, the polynomial calculus, the sequent calculus, the cut-free sequent calculus, as well as the polynomial calculus. Our hardness of approximation results usually apply to proof length measured either by number of symbols or by number of inferences, for tree-like or dag-like proofs. We introduce the Monotone Minimum (Circuit) Satisfying Assignment problem and reduce it to the problems of approximation of the length of proofs.

§1. Introduction. This paper proves lower bounds on the hardness of finding short propositional proofs of a given tautology and on the hardness of finding short resolution refutations. When considering Frege proof systems, which are textbook-style proof systems for propositional logic, the problem can be stated precisely as the following optimization problem:

Minimum Length Frege Proof:

Instance: A propositional formula φ which is a tautology.
Solution: A Frege proof P of φ.
Objective function: The number of symbols in the proof P.

For a fixed Frege system $\mathscr{F}$, let $\min_{\mathscr{F}}(\varphi)$ denote the minimum number of symbols in an $\mathscr{F}$-proof of φ. An algorithm M is said to approximate the Minimum Length Frege Proof problem within a factor of α, if for all tautologies φ, $M(\varphi)$ produces a Frege proof of φ of length $\leq \alpha \cdot \min_{\mathscr{F}}(\varphi)$. (Here, α may be a constant or may be a function of the length of φ.)

We are interested only in *polynomial time* algorithms for solving this problem. However, there is a potential pitfall here since the shortest proof of a propositional

Received July 29, 1998; revised August 3, 1999
The second auhor is supported in part by NSF grants DMS-9503247 and DMS-9803515 and grant INT-9600919/ME-103 from NSF and MŠMT (Czech Republic)
The third author's research is supported by the Bernard Elkin Chair for Computer Science and by US-Israel grant 95-00238
The fourth author is supported in part by NSF grant CCR-9457782, US-Israel BSF grant 95-00238, and grant INT-9600919/ME-103 from NSF and MŠMT (Czech Republic)

0022-4812/00/xxxx-xxxx/$3.10

formula could be substantially longer than the formula itself[1], and in this situation, an algorithm with runtime bounded by a polynomial of the length of the input could not possibly produce a proof of the formula. In addition, it seems reasonable that a "feasible" algorithm which is searching for a proof of a given length ℓ should be allowed runtime polynomial in ℓ, even if the formula to be proved is substantially shorter than ℓ. Therefore we shall only discuss algorithms that are polynomial time in the length of the shortest proof (or refutation) of the input.

Note that an alternative approach would be to consider a similar problem, **Minimum Length Equivalent Frege Proof**, an instance of which is a Frege proof of some tautology φ, and the corresponding solutions are (preferably shorter) proofs of φ. While our results are all stated in terms of finding a short proof to a given tautology, they hold also for that latter version where the instance is a proof rather than a formula.

A yet different approach could be studying algorithms which output the *size* (i.e., number of symbols) of a short proof of the input formula, rather than the proof itself. In this case it is possible for an algorithm to have run time bounded by a polynomial of the length of the input formula, even if the size of the shortest proof is exponential in the size of the formula. In the final section of this paper, we show that strong non-approximability results can be obtained for algorithms with run time bounded by a polynomial of the length of the formula for a variety of proof systems.

A related minimization problem concerns finding the shortest Frege proof when proof length is measured in terms of the number of steps, or lines, in the proof:

Minimum Step-Length Frege Proof:

Instance: A propositional formula φ which is a tautology.
Solution: A Frege proof P of φ.
Objective function: The number of steps in the proof P.

Resolution is a propositional proof system which is popular as a foundation for automated theorem provers. Since one is interested in finding resolution refutations quickly it is interesting to consider the following problem:

Minimum Length Resolution Refutation

Instance: An unsatisfiable set Γ of clauses.
Solution: A resolution refutation R of Γ.
Objective function: The number of inferences (steps) in R.

The main results of this paper state that a variety of minimum propositional proof length problems, including the Minimum Length Frege Proof, the Minimum Step-Length Frege Proof and the Minimum Length Resolution Refutation problems, cannot be approximated to within a factor $2^{\log^{1-o(1)} n}$ by any polynomial time algorithm unless $P = NP$ (we use here the recent result of [15], see Section 2)[2]. Our results apply to all Frege systems, to all extended Frege systems, to resolution, to

[1]Is is known that $NP \neq coNP$ implies that some tautologies require superpolynomially long Frege proofs.

[2]The first version of this paper [1], established somewhat weaker results. Namely, under the assumption of $P \neq NP$, we proved non-approximability to within any constant factor; and under the assumption of $NP \not\subseteq QP$, we proved non-approximability to within a of factor $2^{\log^{1-\varepsilon} n}$. Subsequent to the submission of our paper, [15] has provided improved hardness results for the Minimum Monotone

Horn clause resolution, to the sequent calculus, and to the cut-free sequent calculus; in addition, they apply whether proofs are measured in terms of symbols or in terms of steps (inferences), and they usually apply to either dag-like or tree-like versions of all these systems.

We let $\mathscr{F} \overset{k}{\vdash} \varphi$ mean that φ has an $\mathscr{F}$-proof of $\leq k$ symbols. One of the first prior results about the hardness of finding optimal length of Frege proofs was the second author's result [12] that, for a particular choice of Frege system $\mathscr{F}_1$ with the language $\wedge$, $\vee$, $\neg$ and $\rightarrow$, there is no polynomial time algorithm which, on input a tautology φ and a $k > 0$, can decide whether $\mathscr{F}_1 \overset{k}{\vdash} \varphi$, unless P equals NP. This result however applies only to a particular Frege system, and not to general Frege systems. It also did not imply the hardness of approximating Minimum Length Frege Proofs.

A second related result, which follows from the results of Krajíček and Pudlák [21], is that if the RSA cryptographic protocol is secure, then there is no polynomial time algorithm for approximating the Minimum Step-Length Frege Proof problem to within a polynomial.

Another closely related prior result is the connection between the (non)automatizability of Frege systems and the (non)feasibility of factoring integers that was recently discovered by Bonet-Pitassi-Raz [10]. A proof system T is said be *automatizable* provided there is an algorithm M and a polynomial p such that whenever $T \overset{n}{\vdash} \varphi$ holds, $M(\varphi)$ produces some T-proof of φ in time $p(n)$ (see [13]). Obviously the automatizability of Frege systems is closely related to the solution of the Minimum Length Frege Proof problem: if a proof system S is automatizable, then the minimum length proof problem for S can be approximated to within a polynomial factor. Our theorems give a super-linear lower bound on the automatizability of the Minimum Proof Length problem based on the assumption that $P \neq NP$. It has recently been shown by Bonet-Pitassi-Raz [10] that Frege systems are not automatizable unless integer factorization is efficiently computable, and more recently, that bounded-depth Frege systems are also not automatizable under a similar hardness assumption [8]. The proof of these results actually show that Frege systems are not approximable to within any factor unless integer factorization is sufficiently hard. Thus, they derive stronger non-approximability conclusions than our results, but under a much stronger complexity assumption.

For resolution, the first prior hardness result was Iwama-Miyano's proof in [19] that it is NP-hard to determine whether a set of clauses has a read-once refutation (which is necessarily of linear length). Subsequently, Iwama [18] proved that it is in NP-hard to find shortest resolution refutations; unlike us, he did not obtain an approximation ratio bounded away from 1.

In Section 2, we introduce the MMSA and Circuit MMSA problems and discuss the relevant prior results about the hardness of approximating NP-optimization problems. Section 3 discusses the main results about the hardness of approximating minimum length of refutations on the example of resolution. Section 4 contains the main results about the hardness of approximating minimum length Frege proofs. Section 5 briefly discusses the hardness of approximating shortest sequent calculus

Satisfying Assignment problems (see Theorem 4), and we have revised our paper to incorporate their lower bounds.

proofs and cut-free sequent calculus proofs. Section 6 establishes the hardness of approximating minimum length polynomial calculus proofs. The proofs in Sections 3 through 6 depend critically on the hardness of approximation of (Circuit) MMSA problem.

§**2. Monotone minimum satisfying assignment.** The Monotone Minimum Satisfying Assignment (MMSA) problem is the problem of finding a minimum number of variables in a monotone Boolean formula which need to be set to $\top$ in order to give the formula a true value. This problem was already considered by Goldwasser-Motwani [16, 17] in a very different setting.

This section establishes structural results about the complexity of the MMSA problem from the point of view of the hardness of approximation. For our applications, it is enough to use the recent result of [15] (Theorem 4 below) which shows that MMSA is hard to approximate within $2^{\log^{1-o(1)} n}$ factor unless P=NP. This result appeared after submission of our paper but we prefer to keep the content of this section to give a more global picture.

The reader can find a general introduction to and survey of the hardness of approximation and of probabilistically checkable proofs in [6] and [3]. Recall that an A-reduction, as defined by [20], is a polynomial-time Karp-reduction which preserves the non-approximating ratio to within a constant factor.

Consider the following *NP*-optimization problems:

Monotone Minimum Satisfying Assignment:

Instance: A monotone formula $\varphi(x_1, \dots, x_n)$ over the basis $\{\vee, \wedge\}$
Solution: An assignment $\langle v_1, \dots, v_n \rangle$ such that $\varphi(v_1, \dots, v_n) = \top$.
Objective function: The number of v_i's which equal $\top$.

We henceforth let $\rho(\varphi)$ denote the value of the optimal solution for the MMSA problem for φ i.e., the minimum number of variables v_i which must be set *True* to force φ to have value *True*.

We will also consider the *Circuit MMSA* problem which is to find the minimum number of variables which must be set *True* to force a given monotone circuit over the basis $\{\wedge, \vee\}$ evaluate *True*. It does not matter whether we consider circuits with bounded fanin or unbounded fanin since they can simulate each other. It is apparent that Circuit MMSA is at least as hard as MMSA.

Recall the Minimum Hitting Set problem, which is:

Minimum Hitting Set:

Instance: A finite collection $\mathcal{S}$ of nonempty subsets of a finite set U.
Solution: A subset V of U that intersects every member of $\mathcal{S}$.
Objective function: The cardinality of V.

It is easy to see that MMSA is at least as hard as Minimum Hitting Set: namely Minimum Hitting Set can be reduced (via an A-reduction) to the special case of MMSA where the propositional formula is in conjunctive normal form. Namely, given $\mathcal{S}$ and U, identify members of U with propositional variables and form a CNF formula which has, for each set in $\mathcal{S}$, a conjunct containing exactly the members of that set.

Lund and Yannakakis [22] noted that Minimum Hitting Set is equivalent to Minimum Set Cover (under A-reductions). Furthermore, it follows from [23] that the problem of approximating Minimum Set Cover to within $\Omega(\ln n)$ factor is not in polynomial time unless $P = NP$.

We can get stronger results than the above reduction of Minimum Set Cover to MMSA if we use a construction due to Goldwasser-Motwani [16, 17] and, independently, Arora [private communication] to reduce MMSA to the Minimum Label Cover problem.

Minimum Label Cover: (see [3])

Instance: The input consists of: (i) a regular bipartite graph $G = (U, V, E)$, (ii) an integer N in unary, and (iii) for each edge $e \in E$, a partial function $\Pi_e : \{1, \ldots, N\} \to \{1, \ldots, N\}$ such that 1 is in the range of Π_e.

The integers in $\{1, \ldots, N\}$ are called *labels*. A *labeling* associates a nonempty set of labels with every vertex in U and V. A labeling *covers* an edge $e = (u, v)$ (where $u \in U$, $v \in V$) iff for every label ℓ assigned to v, there is some label t assigned to u such that $\Pi_e(t) = \ell$.

Solution: A labeling which covers all edges.

Objective function: The number of all labels assigned to vertices in U and V.

A Π_4-*formula* is a propositional formula which is written as an AND of OR's of AND's of OR's.

Theorem 1 (Goldwasser-Motwani, Arora). *There is an A-reduction from Minimum Label Cover to MMSA such that the instances of Label Cover are mapped to Π_4 formulas.*

Proof. Suppose we have an instance of Label Cover $(G, N, \{\Pi_e\})$. Let $m = |U|$ and $k = |V|$. For $1 \leq i \leq m$ and $1 \leq \ell \leq N$, let u_i^ℓ be a propositional variable with the intended meaning that $u_i^\ell = \top$ iff ℓ is one of the labels assigned to vertex $i \in U$. Likewise, for $1 \leq j \leq m$ and $1 \leq \ell \leq N$ let v_j^ℓ denote the condition that ℓ is one of the labels assigned to vertex $j \in V$. We shall construct a formula φ involving the variables u_i^ℓ and v_j^ℓ so that the minimal satisfying assignments for φ are precisely those truth assignments which correspond to minimal weight labelings which cover all edges. We define φ to be

$$\bigwedge_{j=1}^{k} \bigvee_{\ell=1}^{N} \left(v_j^\ell \wedge \bigwedge_{i|(i,j)\in E} \; \bigvee_{t|\Pi_e(t)=\ell} u_i^t \right)$$

The formula φ clearly is a monotone Π_4-formula and has size polynomial in N, m, k. It is easy to verify that any minimum satisfying assignment for φ corresponds to a labeling which covers all edges and which has a minimum number of labels: to see this, one should note that any minimum size labeling, as well as an minimum satisfying assignment, will have exactly one label assigned to each vertex V. ⊣

It was proved in [2] that Minimum Label Cover is not approximable within a $2^{\log^{(1-\varepsilon)} n}$ factor unless $NP \subseteq QP$. An immediate corollary of Theorem 1 is that MMSA enjoys the same hardness of approximation, even when restricted to Π_4-formulas.

We next present a second proof that MMSA is hard to approximate to within a $2^{\log^{(1-\varepsilon)} n}$ factor; although it does not yield the hardness of the Π_4-formula case, it may be independent interest since because of its use of "self-improvement" and of non-constant depth propositional formulas.

LEMMA 2 (Self-improvement property). *For any formulas φ_1, φ_2 there is a formula $\varphi_1 * \varphi_2$ such that $\rho(\varphi_1 * \varphi_2) = \rho(\varphi_1) \cdot \rho(\varphi_2)$ and $|\varphi_1 * \varphi_2| \leq |\varphi_1| \cdot |\varphi_2|$.*

PROOF. Let $\varphi_1 = \varphi_1(x_1, \ldots, x_k)$, $\varphi_2 = \varphi_2(x_1, \ldots, x_\ell)$. Let $(\varphi_1 * \varphi_2)$ be a composition of φ_1 and φ_2:

$$(\varphi_1 * \varphi_2)(x_1^1, \ldots, x_1^\ell, \ldots, x_k^1, \ldots, x_k^\ell) = \varphi_1(\varphi_2(x_1^1, \ldots, x_1^\ell), \ldots, \varphi_2(x_k^1, \ldots, x_k^\ell))$$

The proof follows. ⊣

THEOREM 3. *There is no polynomial time algorithm which approximates MMSA within a factor of $2^{(\log n)^{1-\varepsilon}}$ for any $\varepsilon > 0$ unless $NP \subseteq QP$.*

PROOF. Suppose we have a formula ψ which is an instance of Satisfiability. Let $n = |\psi|$. By the reduction to Minimum Hitting Set, there exists a polynomial time reduction from Satisfiability to approximation of MMSA within factor 2 [7], i.e., a function $f : \psi \mapsto \varphi, |\varphi| = n^{O(1)}$ such that if ψ is satisfiable then $\rho(f(\psi)) < C_1(n)$ and otherwise $\rho(f(\psi)) > C_2(n)$ with gap $g(n) = \frac{C_2(n)}{C_1(n)} > 2$.

Applying self-improvement k-times to φ, we increase the gap g to 2^k. This gives the formula $F(\psi) = f(\psi) * f(\psi) * \ldots * f(\psi)$ (k-times), which has length $N = |F(\psi)| < |\varphi|^k$. If ψ is satisfiable than $\rho(F(\psi)) < C_1^k(n)$ otherwise $\rho(F(\psi)) > C_2^k(n)$, the new gap is at least 2^k.

If we take $k(n) = (\log n)^c$, we get

$$N = |\varphi|^{k(n)} \leq n^{(\log n)^c}, \quad \text{and} \quad \log n \geq (\log N)^{\frac{1}{c+1}},$$

so,

$$g \geq 2^{k(n)} \geq 2^{(\log N)^{\frac{c}{c+1}}}.$$

Suppose we have an algorithm A which approximates MMSA to within the factor $2^{(\log n)^{1-\varepsilon}}$. Take c large enough so that that $c/(c+1) > 1 - \varepsilon$. The function F is computable in QP, and applying algorithm A to the formula $F(\psi)$ determines the satisfiability of ψ.

That completes the proof of Theorem 3. ⊣

Recently [15] improved both the factor and the hypothesis of Label Cover. They considered the case of MMSA for Π_3-formulas and showed its hardness directly from a strong version of PCP-Theorem. They also show how to reverse the reduction of Theorem 1 and reduce MMSA for Π_3-formulas to Label Cover.

We will use their result in further sections:

THEOREM 4 (I. Dinur, S. Safra, [15]). *If $P \neq NP$, then there is no polynomial time algorithm which can approximate MMSA (and hence Circuit MMSA) within a factor $2^{\log^{1-o(1)} n}$.*

In the next sections we reduce the Circuit MMSA problem first to the Minimum Length Resolution Refutation problem and then to other problems on minimum proof length. This will establish the same hardness of approximation results for these proof length optimization problems.

§3. The hardness of refutations. In this section we prove the simplest hardness result for resolution. The general idea of the proof can be applied to any "reasonable" refutation system, including natural systems as the polynomial calculus, Frege systems, bounded depth Frege systems, sequent calculi, and cut-free sequent calculi.

Resolution is a well-known proof system and its extensions are widely used as a foundation for many theorem proving systems. Thus it is of particular interest that it is difficult to find approximately shortest resolution refutations.

We shall be concerned exclusively with propositional resolution systems. Recall that a *literal* is either a variable p or the negation of a variable $\overline{p}$. A *clause* is a finite set of literals and is interpreted as the disjunction of its members. A *set of clauses* Γ is interpreted as the conjunction of its member clauses; thus a set of clauses can be identified with a formula in conjunctive normal form.

The resolution rule allows an inference of the form

$$\frac{C \cup \{p\} \qquad D \cup \{\overline{p}\}}{C \cup D}$$

It is well-known that resolution is sound and complete as a refutation system; namely, a set Γ is unsatisfiable if and only if the empty clause can be derived using only resolution inferences from the clauses in Γ.

A *Horn* clause is a clause which contains at most one positive literal.

A resolution refutation consists of a sequence of clauses, ending with the empty clause. We can measure the length or size of a refutation in terms of either its step-length or its symbol-length. The *step-length* of the refutation is just equal to the number of clauses in the refutation. The *symbol-length* is defined to equal the sum of the cardinalities of the clauses appearing in the refutation. (There seems to be no fixed convention on how to measure the length of resolution refutations: thus, we shall always explicitly include one of the modifiers 'step-' or 'symbol-'.)

A refutation can be either tree-like or dag-like: unless it is explicitly stated otherwise, refutations are considered to be dag-like. We shall obtain the best possible results in that our upper bounds on length will apply to tree-like refutations and our lower bounds on length will apply to dag-like refutations. In the introduction, we introduced the Minimum Length Resolution Refutation problem: henceforth we'll be more precise and talk about the Minimum Step-Length Resolution Refutation and the the Minimum Symbol-Length Resolution Refutation problems.

THEOREM 5. *There is an A-reduction from the Circuit MMSA problem to the Minimum Length Resolution Refutation problems. This reduction works for both tree-like and dag-like refutations and for both step-length and symbol-length. Furthermore, the reduction produces only sets of Horn clauses.*

Together with Theorem 4 this yields

COROLLARY 6. *If $P \neq NP$, then there is no polynomial time algorithm which can approximate Minimum Step-Length (Symbol-Length) Resolution Refutation to within $2^{\log^{1-o(1)} n}$ factor.*

These hardness results apply to both dag-like and tree-like resolution. In addition, the hardness results apply to Horn resolution, where all the input clauses are Horn clauses.

To prove Theorem 5, we shall construct the reduction from Circuit MMSA to Minimum Length Resolution Refutation problems. Let C be an instance of Circuit MMSA; we define a set of clauses Γ_C which will be an A-reduction to the Minimum Length Resolution Refutation problems.

Enumerate the subcircuits of C as $C_1, \dots, C_\ell$, where the input variables are first in the enumeration and where each C_i is listed only after all of its own subcircuits are enumerated, and thus C_ℓ is C. Obviously the number ℓ of subcircuits is less than the number of symbols n in C. We introduce new propositional variables $y_1, \dots, y_\ell$, and define the set Γ_C to contain the following clauses:

a. The clause $\{\overline{y_\ell}\}$ is in Γ_C.

b. For each $i \le \ell$, if C_i is $(C_j \wedge C_k)$, then the clause $\{\overline{y_j}, \overline{y_k}, y_i\}$ is in Γ_C.

c. For each $i \le \ell$, if C_i is $(C_j \vee C_k)$, then the clauses $\{\overline{y_j}, y_i\}$ and $\{\overline{y_k}, y_i\}$ are in Γ_C.

The above clauses describe the evaluation of C; however, note that they say nothing about the truth of the input variables $x_1, \dots, x_p$ of C. For each variable x_i of C, we introduce new variables $x_{i,j}$ for $j = 1, 2, \dots, m$, and further include in Γ_C the following clauses:

d. For each $i \le p$, the clauses $\{x_{i,1}\}$ and $\{\overline{x_{i,m}}, y_i\}$ and

$$\{\overline{x_{i,j}}, x_{i,j+1}\} \quad \text{for } j = 1, \dots, m-1$$

are included in Γ_C. These clauses are said to be *associated with* x_i.

That completes the definition of Γ_C. Informally, Γ_C asserts that there exists a truth-evaluation to all subcircuits of C such that: the truth evaluation given to the output circuit, y_ℓ, is 0, the truth evaluation given to all input variables y_i $(i \le p)$ is 1, and the truth evaluation is consistent with all intermediate gate values. Clearly, this is an unsatisfiable formula since C is monotone.

The purpose of the clauses in d. is to force any derivation of an input y_i to require m steps. Our goal is to show that the price to infer the literal y_ℓ, and hence get a contradiction, is nearly equal to ρ, the size of a minimal satisfying assignment for C, multiplied by the price to infer any of the "input" literals y_i, $i \le p$.

Lemma 7. *Let C be an instance of Circuit MMSA and let ρ equal the cardinality of the minimum satisfying assignment for C.*

(1) *Γ_C has a dag-like refutation with symbol-length (and hence step-length) equal to $O(\rho m + n)$.*

(2) *Γ_C has a tree-like refutation of symbol-length $O(\rho m + n^2)$ and step-length $O(\rho m + n)$.*

Proof. (a) Let $I \subseteq \{x_1, \dots, x_p\}$ specify a satisfying assignment for C of cardinality ρ. The dag-like proof proceeds by first deriving the clause $\{y_i\}$ for each $x_i \in I$. Each $\{y_i\}$ is derived in m steps using the clauses associated with x_i; this part of the refutation takes ρm steps. The refutation then derives clauses $\{y_i\}$ starting with smaller values of i and ending with $\{y_\ell\}$ (e.g., the subcircuits of C are processed in a bottom-up order). This takes $O(n)$ steps. One further resolution with the input clause $\{\overline{y_\ell}\}$ completes the refutation. Each clause in the refutation contains a constant number of (in fact, at most three) literals. Hence the symbol-length of the refutation is also $O(\rho m + n)$.

(b) The above proof is clearly not tree-like. To form a tree-like refutation, we use a top-down procedure to generate the refutation. The first phase of the refutation starts with the clause $\{\overline{y_\ell}\}$ and derives successively clauses of the form $\{\overline{y_{k_1}}, \overline{y_{k_2}}, \ldots, \overline{y_{k_r}}\}$ with $k_1 > k_2 > \cdots > k_r$. Such a clause is resolved with one of the (at most two) clauses that contain y_{k_1} positively. This continues until we have a clause which contains only literals $\overline{y_i}$ corresponding to input x_i of C. It is possible to do this so that the remaining clause is just $\{\overline{y_i} : x_i \in I\}$. For the second phase of the refutation, derive the clauses $\{y_i\}$, for $x_i \in I$, with ρm steps, and for the third phases, use ρ resolutions to derive the empty clause.

There are obviously $O(n)$ steps in the first and third phases of the derivation, so the whole refutation has $O(\rho m + n)$ steps. Furthermore, each clause in the first and third phase contains at most n literals. The second phase contains ρm clauses each with a constant number of literals. Therefore, the symbol-length of the tree-like refutation is $O(\rho m + n^2)$. ⊣

LEMMA 8. *Let C and ρ be as above. Then any resolution refutation (dag-like or tree-like) must have step-length, and hence symbol-length, of at least ρm.*

PROOF. Let R be a resolution refutation. An input variable x_i is defined to be R-*analyzed* if every one of the $(m+1)$-clauses associated with x_i is used in the refutation R. Obviously it will suffice to prove that at least ρ input variables are R-analyzed. In fact, if I is defined to equal the set of R-analyzed variables, then I implies a satisfying assignment for C.

This last fact is almost immediate. To prove it formally, we define a truth assignment τ as follows: (1) τ assigns truth values to variables y_i according to the value I assigns to C_i (2) τ assigns *True* to $x_{i,k}$ iff each clause $\{x_{i,1}\}$ and $\{\overline{x_{i,j}}, x_{i,j+1}\}$ for $1 \leq j < k$ is used in R. If I doesn't satisfy C, then τ would satisfy all the clauses used in the refutation R, which is impossible. Therefore, I is a satisfying assignment for C. ⊣

Let us choose m sufficiently large with respect to n^2 say $m = n^3$. These two Lemmas establish that $C \mapsto \Gamma_C$ is an A-reduction; namely, it is easy to see that the constructed reduction transforms a sufficiently large gap $g(n)$ between hard and easy instances of Circuit MMSA into a gap $g_\varepsilon(n)$ in Minimum Length Resolution Refutation, hence if Minimum Length Resolution Refutation is approximable with some factor $f(n)$ then Circuit MMSA is approximable with $O(f(n^{O(1)}))$. This proves Theorem 5.

§4. **Main results for Frege systems.** In this section we prove the existence of an A-reduction from the Circuit MMSA to the Minimum (Step) Length Frege Proof problem. We prove this for both tree-like and dag-like Frege proofs. This will imply the hardness of approximation of Minimum (Step) Length Frege Proof within a factor of $2^{\log^{1-o(1)} n}$.

4.1. Preliminaries. Frege proof systems are proof systems for propositional logic. A Frege proof system $\mathscr{F}$ is specified by its language L and a finite set of inference and axiom schemes. The language L is a finite set of Boolean connectives, which is complete in the sense that any Boolean function can be represented by an L-formula. The permissible inferences are specified schematically as inferences

$$\frac{A_1 \quad A_2 \ \cdots \ A_k}{B}$$

which indicates that for any substitution σ of formulas for variables, $B\sigma$ may be inferred from the formulas $A_1\sigma, \ldots, A_k\sigma$. We allow $k = 0$ in the above scheme, which corresponds to axioms. Finally, the Frege proof system must be implicationally complete, i.e., if $A_1, \ldots, A_k \models B$ then there is a derivation of B from the assumptions $A_1, \ldots, A_k$ using the inferences of $\mathscr{F}$.

We define the size or length, $|C|$, of a formula C to equal the number of symbols in C, where each occurence of a variable or a connective is counted as a symbol. Likewise, if P is a Frege proof, then the symbol size of P, $|P|$, equals the number of symbols in P. If $\mathscr{F}$ is a Frege system, then $\mathscr{F} \vdash \varphi$ means that there is an $\mathscr{F}$-proof of φ. The *symbol size* of a Frege proof is the total number of symbols in the proof. The *step-length* (or length) of a Frege proof is the number of lines in the proof. $\mathscr{F} \overset{n}{\vdash} \varphi$ means that there exists an $\mathscr{F}$-proof P such that $|P| \leq n$.

Typical examples of Frege system include the 'textbook systems' which use the language $\{\wedge, \vee, \neg, \rightarrow\}$ and have a finite set of axiom schemes and have modus ponens as their only other rule of inference. Of course there are many possible such textbook systems since there are many choices for the axiom schemes; however, they are all essentially equivalent in terms of proof length. Indeed the following holds:

THEOREM 9 ([14, 24, 25]). *If $\mathscr{F}_1$ and $\mathscr{F}_2$ are Frege systems with the same language, then they linearly simulate each other; i.e., for all φ, if $\mathscr{F}_1 \overset{n}{\vdash} \varphi$ then $\mathscr{F}_2 \overset{O(n)}{\vdash} \varphi$, and vice-versa.*

For Frege proof systems in differing languages, it is known that any two Frege systems $\mathscr{F}_1$ and $\mathscr{F}_2$ p-simulate each other, i.e., that any $\mathscr{F}_1$-proof can be translated into an $\mathscr{F}_2$-proof in polynomial time and vice-versa; see [14, 24] for precise definitions and proofs of this.

Extended Frege proof systems are propositional proof systems which allow the introduction of abbreviations of formulas on the fly. It is conjectured that the minimal symbol size for extended Frege proofs can sometimes be exponentially smaller than the corresponding minimal Frege proof; however, this is still open.

We next define the notion of "active" formulas in a proof, which will be useful for proving lower bounds on the lengths of proofs. Recall that an inference in a proof must be a substitution instance of an axiom scheme, i.e., each inference must be of the form

$$\frac{A_1\sigma \ \cdots \ A_k\sigma}{A_{k+1}\sigma} \ (\mathrm{I})$$

Consider a particular occurrence of a formula C as a subformula of a formula $A_i\sigma$ in the inference (I). If the principal connective of C is present already in the formula A_i, then we say C is *active* w.r.t. the inference (I). Otherwise, C occurs as a (not necessarily proper) subformula of $x\sigma$ for some variable x, and C is not active w.r.t. inference (I).

If a formula C has some occurrence in a proof P which is active with respect to some inference of P, then C is said to be *active* in P. (The terminology is potentially confusing: it is important to note that an active formula of P may never occur as a formula in the proof P, but instead only as a subformula of formulas in P.)

The next theorem lets us obtain a lower bound on the length of P, $|P|$, in terms of the lengths of the active formulas of P.

THEOREM 10 (see [11]). *Let $\mathscr{F}$ be a Frege proof system. There is a constant ε such that if P is a Frege proof and we let φ range over active formula-occurrences in P, then*

$$|P| \ \geq \ \varepsilon \cdot \sum_{\varphi} |\varphi|$$

PROOF. A formula can be viewed as a tree with nodes labeled by connectives from L. The *depth* of a formula is defined to equal the height of this tree, namely the maximum number of connectives along any branch of the tree. Let d equal the maximum depth of the depths of the formulas which occur in the inference schemes of $\mathscr{F}$, and set $\varepsilon = (1/d)$. Clearly, any active occurrence of a formula in P must have its principal connective at distance at $d - 1$ from the root of the formula's tree. Thus, any given single symbol a occurring in P can occur inside at most d active occurrences of subformulas. From this, we immediately get

$$d \cdot |P| \ \geq \ \sum_{\varphi} |\varphi|$$

and the theorem follows immediately. ⊣

As mentioned earlier, the *step-length* of a proof P is equal to the number of steps or inferences (counting axioms as nullary inferences) in the proof. There is a linear relationship between the number of formulas active in P and the step length of P; namely,

THEOREM 11. *Let $\mathscr{F}$ be a Frege proof system. Then there is a constant ε so that if P is a proof and m is the number of distinct active formulas in P, then the step-length of P is $\geq \varepsilon m$.*

PROOF. We let α equal the maximum number of subformulas that can be active in any given formula in P. For instance, if every inference scheme has depth bounded by d and r is the maximum arity of connectives in the language of F, then $\alpha = \sum_{i=0}^{d} r^i$ works. Clearly Theorem 11 is true with $\varepsilon = 1/\alpha$. ⊣

It is an interesting (albeit trivial) observation that if no formula is repeated in P, then the number of steps in P is also linearly upper bounded by the number of distinct formulas active in P.

4.2. Hardness of approximation for Frege systems.

THEOREM 12. *There is an A-reduction from the Circuit MMSA problem to the Minimum (Step) Length Frege Proof problems. This reduction works for both tree-like and dag-like inferences.*

Together with Theorem 4 this yields

Corollary 13. *If $P \neq NP$, then there is no polynomial time algorithm which can approximate Minimum* (*Step*) *Length Frege Proof to within $2^{\log^{1-o(1)} n}$ factor.*
These hardness results apply to both dag-like and tree-like Frege Systems.

The outline of our proof is as follows: Suppose we have a monotone circuit C with inputs $x_1, \dots, x_k$ given as a set of instructions $x_i := x_{j_{i,1}} OP x_{j_{i,2}}$ for $i \in \{k+1, \dots, n\}$, where OP is $\wedge$ or $\vee$, where $j_{i,1}, j_{i,2} < i$ and where x_n is the output node. We construct a tautology

$$\psi_C = \left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} OP x_{j_{i,2}}) \to x_i) \right) \to x_n,$$

where $x_1, \dots, x_k$ are replaced with some hard "independent" tautologies $\tau_1, \dots, \tau_k$. Then we claim that the complexity of a proof of ψ_C is about $\rho(C)$ multiplied by the complexity of a single "hard" tautology τ_i (the intuition is that we need need a lengthy proof to establish $\rho(C)$ many tautologies from among $\tau_1, \dots, \tau_k$ which force C to true, and then we need "few" steps to infer ψ_C by evaluation of our circuit).

We use the construction of [12] to define this set of hard tautologies

Definition. Let $p_0, p_1, \dots$ be propositional variables. Let $\perp$ be the contradiction $p_0 \wedge \neg p_0$, and let τ_i^0 be the tautology $(\neg p_i \vee p_i)$. Define

$$\tau_i^\ell = \underbrace{(\perp \vee (\perp \vee \cdots (\perp \vee \tau_i^0) \cdots))}_{\ell \ times}$$

Note that $|\tau_i^l| = O(\ell)$.

We wish to substitute formulas τ_i^m into ψ_C, but there is the problem that the Frege system's language may not contain the connectives $\vee, \wedge, \neg$, so the τ_i^m may not be well-formed formulas. Let us fix some Frege proof system $\mathscr{F}$, with language L. We wish to construct L-formulas $\tau_i^{\ell,L}$ which are analogues of the formulas τ_i^ℓ. To do this with similar size bounds, we need the following lemma:

Lemma 14 (Reckhow [24]). *There are L-formulas $NOT(x, z)$, $AND(x, y, z)$, $OR(x, y, z)$, and $IMP(x, y, z)$ such that*

(1) *$NOT(x, z)$ contains one occurrence of x, and $AND(x, y, z)$, $OR(x, y, z)$, and $IMP(x, y, z)$ contain exactly one occurrence of each of x and y.*

(2) *The four formulas represent the Boolean functions $\neg x$, $(x \wedge y)$, $(x \vee y)$ and $(x \to y)$; in particular, the truth values of the formulas are independent of the truth value of z.*

Proof. The side variable z acts merely as a placeholder whose truth value is irrelevant: in fact, if the language L contains a constant symbol, then the use of the variable z is unnecessary. We shall assume that the constant symbols $\top$ and $\perp$ are included in L; this may be assumed without loss of generality since the symbols $\top$ and $\perp$ may be replaced everywhere by L-formulas equivalent to the formulas $(z \vee \neg z)$ and $(z \wedge \neg z)$, respectively.

Let N be an L-formula containing only the variable x which represents the propositional function $\neg x$; N exists since L is a complete set of connectives. If N contains n occurrences of x, we write $N = N(x, x, \dots, x)$ with each occurrence of x separately indicated. Let $\top^i$ denote a vector of i occurrences of $\top$, and define $\bot^i$ similarly. By choice of N, $N(\top^n)$ has value *False* and $N(\bot^n)$ has value *True*. Therefore, there is some $0 \leq i < n$ such that $N(\top^i, \bot^{n-i})$ has value *True* and $N(\top^{i+1}, \bot^{n-i-1})$ has value *False*. Thus, we can take $NOT(x)$ to be the formula $N(\top^i, x, \bot^{n-i-1})$.

To prove the remainder of the lemma, it will suffice to find an L-formula $X(x, y)$ which has one occurrence of each of x and y, and which has appearing in its truth table either three values *True* and one value *False*, or three values *False* and one value *True*. (Note that conjunction, disjunction and implication are three of the eight propositional functions whose truth table has this property.) This will suffice since AND, OR and IMP can be readily defined from such a formula X and from NOT.

Let A be an L-formula containing only the variables x and y which represents the propositional function $(x \wedge y)$. Assume $A = A(x, \dots, x, y, \dots, y)$ has m occurrences of x and n occurrences of y, each occurrence separately indicated. Define $A_{i,\top}$ to be the formula $A(\top^i, \bot^{m-i}, \top^n)$ and $A_{i,\bot}$ to be $A(\top^i, \bot^{m-i}, \bot^n)$. Now $A_{m,\top}$ and $A_{m,\bot}$ have different truth values, and $A_{0,\top}$ and $A_{0,\bot}$ have the same truth value. Clearly, there is a value $0 \leq i < m$ so that $A_{i,\top}$ and $A_{i,\bot}$ have the same truth value, but so that $A_{i+1,\top}$ and $A_{i+1,\bot}$ have different truth values. Fix such an i and let $B = B(x, y, \dots, y)$ be the formula $A(\top^i, x, \bot^{m-i-1}, y, \dots, y)$. Note that B has one occurrence of x and n occurrences of y, each indicated separately. Let $B_i(x)$ be the formula $B(x, \top^i, \bot^{n-i})$. From the definition of B, the multiset of the four truth values of $B_0(\top)$, $B_0(\bot)$, $B_n(\top)$ and $B_n(\bot)$ contains either three *Trues* and one *False*, or one *True* and three *Falses*. Therefore, there is some value $0 \leq j < n$ so that the multiset of the truth values of $B_j(\top)$, $B_j(\bot)$, $B_{j+1}(\top)$ and $B_{j+1}(\bot)$ enjoys the same property. Letting $X(x, y)$ be the formula $B(x, \top^j, y, \bot^{n-j-1})$ gives the desired formula. ⊣

For simplicity of notation, we shall henceforth suppress mentioning the occurrences of the side variable z.

DEFINITION. If φ is a formula over the basis $\{\neg, \wedge, \vee, \rightarrow\}$, then its *$L$-translation*, φ^L, is the L-formula obtained by replacing the connectives $\neg$, $\wedge$, $\vee$, and $\rightarrow$ with the formulas NOT, AND, OR and IMP in the obvious way. Because of the condition that x and y occur at most once in the formulas NOT, AND, OR and IMP, the size $|\varphi^L|$ of φ^L is $O(|\varphi|)$.

We write $\tau_i^{\ell,L}$ to denote $(\tau_i^\ell)^L$; thus $|\tau_i^{\ell,L}| = O(\ell)$.

The next lemma will be used to give upper bounds on the lengths of $\mathscr{F}$-proofs.

LEMMA 15 ([12]). *$\tau_i^{\ell,L}$ has an $\mathscr{F}$-proof of length $O(\ell^2)$ (step-length $O(\ell)$).*

This lemma is proved by noting that $\mathscr{F}$ can derive successively $\tau_i^{0,L}$, $\tau_i^{1,L}$, $\tau_i^{2,L}$, etc., until $\tau_i^{\ell,L}$ is derived. ⊣

Suppose that we are given the circuit C. Let us take the formula ψ_C defined in the begining and translate to our language:

$$\psi_C^L = \left[\left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} OP x_{j_{i,2}}) \rightarrow x_i) \right) \rightarrow x_n \right]^L,$$

where $x_1, \dots, x_k$ are replaced with $\tau_1^{m,L}, \dots, \tau_k^{m,L}$ for sufficiently large m (it will be enough to let $m = n^3$). We claim that ψ_C^L is the reduction of Circuit MMSA instance C to Minimum (Step) Length Frege Proof problem. We are going to show that minimal proof length of ψ_C^L is about $\rho(C) \cdot m^2$ (step-length of ψ_C^L is about $\rho(C) \cdot m$).

Lemma 16. *ψ_C^L has a tree-like $\mathscr{F}$-proof of length $O(\rho(C) \cdot m^2 + m \cdot n^2 \log n)$ (step-length $O(\rho(C) \cdot m + n \log n)$).*

Proof. Suppose that $\langle v_j \rangle$ is the minimal satisfying assignment for C and that I is the index set of all v_i such that $v_i = \top$, $|I| = \rho(C)$. First we infer all the tautologies $\tau_i^{m,L}$, $i \in I$ in length $\rho(C) \cdot m^2$ (step length $\rho(C) \cdot m$) by Lemma 15.

Let $r_1, r_2, \dots, r_s = n$ be the increasing sequence of indices such that $x_{r_1}, x_{r_2}, \dots, x_{r_s}$ are made true by the truth assignment $\vec{v}$. Then, it is straightforward to construct a tree-like Frege proof of the formulas

$$\psi_\ell = \left[\left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} OP x_{j_{i,2}}) \to x_i) \right) \to \bigwedge_{i=1}^{\ell} x_{r_i} \right]^L$$

which proceeds by proving these successively with $\ell = 1, 2, 3, \dots, s$ using $\tau_i^{m,L}$, $i \in I$ as basis. To make our inference tree-like on each step ℓ we independently prove formulas

$$\left[\left(\bigwedge_{i=k+1}^{n} ((x_{j_{i,1}} OP x_{j_{i,2}}) \to x_i) \right) \to ((x_{j_{r_{\ell+1},1}} OP x_{j_{r_{\ell+1},2}}) \to x_{r_{\ell+1}}) \right]^L,$$

$$\left[\left(\bigwedge_{i=1}^{\ell} x_{r_i} \right) \to (x_{j_{r_{\ell+1},1}} OP x_{j_{r_{\ell+1},2}}) \right]^L$$

Together with ψ_ℓ it will infer $\psi_{\ell+1}$ in $O(1)$ steps. Since the conjunctions all have $O(n)$ inputs and since the formulas have symbol-length $O(m \cdot n)$ if one is careful and uses balanced conjunctions [9], the inference of $\psi_{\ell+1}$ from ψ_ℓ can have length $O(m \cdot n \log n)$ (step-length $O(\log n)$).

Finally the overall proof has length at most $O(\rho(C) \cdot m^2 + m \cdot n^2 \log n)$ (step-length $O(\rho(C) \cdot m + n \log n)$). Lemma 16 follows. ⊣

Definition. Let ψ be a formula and consider a particular $\tau_i^{\ell,L}$. We define $\psi/(\tau_i^{\ell,L})$ to be the formula obtained from ψ by replacing every occurrence of $\tau_i^{\ell,L}$ as a subformula of ψ with the formula $\bot$. Note, that if $\ell_1 < \ell_2$ then $\tau_i^{\ell_2,L}/(\tau_i^{\ell_1,L})$ is not a tautology anymore.

If P is a proof, then we define $P/(\tau_i^{\ell,L})$ be a sequence of formulas obtained by replacing every ψ in P with $\psi/(\tau_i^{\ell,L})$. Note that $P/(\tau_i^{\ell,L})$ will not, in general, be a valid proof.

Lemma 17. *Let P be a proof of ψ and suppose that the formula $\tau_i^{\ell,L}$ is not active in P. Then, $\psi/(\tau_i^{\ell,L})$ is a tautology.*

The proof of Lemma 17 is immediate by the fact that if $\tau_i^{\ell,L}$ is not active in P, then $P/(\tau_i^{\ell,L})$ is identical to P, except that it may use a different substitution of formulas for variables, and hence it is still a valid proof. ⊣

LEMMA 18. *Any $\mathscr{F}$-proof of ψ_C^L has length $\Omega(\rho(C)\cdot m^2)$ (step-length $\Omega(\rho(C)\cdot m)$).*

PROOF. Suppose that $\rho(C) = p$. Let P be some $\mathscr{F}$-proof of ψ_C^L. Let I be the index set of all i such that $\tau_i^{\ell,L}$ is active in P for all $0 \leq \ell \leq m$. For $j \notin I$, choose j_r so that $\tau_j^{j_r,L}$ is not active in P. By Lemma 17 we have that, after replacing all $\tau_j^{j_r,L}$ with $\perp$ for all $j \notin I$, the formula ψ_C^L remains a tautology. Hence the circuit C is satisfied by the truth assignment corresponding to the characteristic function of I, hence $|I| \geq p$. Thus $|P| = \Omega(p \cdot m^2)$ by Theorem 10 and the fact that the total length of the formulas $\tau_i^{\ell,L}$, for $i \in I,\ 0 \leq \ell \leq m$, is $\Omega(p \cdot m^2)$. Analogously by Theorem 11 the step length of P is $\Omega(p \cdot m)$. ⊣

Altogether Lemmas 16, 18 imply that the mapping $C \mapsto \psi_C^L$ is A-reduction. Theorem 12 follows.

REMARK. All of our hardness results for approximating step-length and symbol-length of Frege proofs also apply to extended Frege systems. To see this, it suffices to note that all the upper and lower bounds on the length of Frege proofs which were obtained in the proof of Theorem 12, also apply to extended Frege proofs. Of course it is obvious that the upper bounds apply since every Frege proof is an extended Frege proof. The lower bounds also apply, since Theorems 10 and 11 are also true for extended Frege systems ([11]).

§5. The propositional sequent calculus. This section covers the hardness results for the propositional sequent calculus: somewhat surprisingly, the hardness results apply equally to the sequent calculus with cuts and to the cut-free sequent calculus. We presume the reader is familiar with the sequent calculus: any of the usual variants of the sequent calculus may be used with the proviso that initial sequents are of the form $A \to A$ where A may be *any* formula (not necessarily atomic).

The propositional sequent calculus (with cuts allowed) and Frege systems are very close in strength and are known to p-simulate each other (actually they simulate each other to within a factor of $O(\log n)$, see [9]).

THEOREM 19. *There is an A-reduction from the Circuit MMSA problem to the Minimum Length Propositional Sequent Calculus Proof problem.*

As usual, this theorem holds for proof length measured in terms of either number of symbols or number of steps. In addition, it holds for the tree-like and the dag-like versions of the sequent calculus. Since the proof of Theorem 19 is quite similar to the proofs of Section 4, we shall omit it.

As an immediate corollary, the propositional sequent calculus enjoys the same hardness results as we have obtained for the other proof systems.

We now turn to the cut-free propositional sequent calculus. Our main theorem implies that all of our hardness results apply also to this proof system:

THEOREM 20. *There is an A-reduction from the Circuit MMSA problem to the Minimum Length Cut-Free Propositional Sequent Calculus Proof problem.*

For the proof of this theorem, we will presume that the sequent calculus includes the connectives $\wedge$, $\vee$ and $\to$ (although the results hold even if the only connectives are $\wedge$ and $\vee$). We use $\Rightarrow$ for the sequent connective (which should not be confused with the implication sign $\rightarrow$!) We remind the reader of the Kreisel-Takeuti trick of replacing cuts with $\rightarrow$:left inferences:

$$\frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \qquad \Rightarrow \qquad \frac{\Gamma \Rightarrow \Delta, A \quad A, \Gamma \Rightarrow \Delta}{A \rightarrow A, \Gamma \Rightarrow \Delta}$$

PROOF. Define the formulas τ_i^m as in Section 4.2. As before, assume we have a circuit C with inputs $x_1, \ldots, x_k$ and with internal gates $x_{k+1}, \ldots, x_n$ which is specified as a set of instructions $x_i := x_{j_{i,1}} OP_i x_{j_{i,2}}$ where each OP_i is $\wedge$ or $\vee$ and where $j_{i,1}, j_{i,2} < i$. The gate x_n is the output of C. Let Δ be the cedent containing the following formulas:

a. If OP_i is $\wedge$, then Δ contains the formula $x_{j_{i,1}} \wedge x_{j_{i,2}} \rightarrow x_i$.
b. If OP_i is $\vee$, then Δ contains the two formulas $x_{j_{i,1}} \rightarrow x_i$ and $x_{j_{i,2}} \rightarrow x_i$.

Let the formula $\chi(x_1, \ldots, x_n)$ be

$$\left(\bigwedge_{i=1}^{n} (x_i \rightarrow x_i) \wedge \bigwedge \Delta \right) \rightarrow x_n.$$

Let $\psi = \psi(x_{k+1}, \ldots, x_n)$ be the formula obtained from χ by replacing each variable x_i with $i \leq k$ with τ_i^m.

Using the 'active formulas' theorems (Theorems 10 and 11), any proof of ψ must have step-length at least $\rho \cdot m$ and must have symbol-length at least $\rho \cdot m^2$. As usual, this lower bound applies to either dag-like or tree-like proofs.

Now we prove there is a tree-like cut-free proof of ψ which has step-length $O(\rho \cdot m + n^2)$ and symbol-length $O(\rho \cdot m^2 + m \cdot n^3)$. Taking $m = n^4$, this will complete the proof of Theorem 20.

Let v be a minimum satisfying assignment and $I = \{i : v(x_i) = \top\}$. So $|I| = \rho$. We let T_I^m be the cedent

$$\tau_{i_1}^m, \tau_{i_2}^m, \ldots, \tau_{i_\rho}^m,$$

where $I = \{i_1, \ldots, i_\rho\}$. Let Γ^m denote the cedent

$$\tau_1^m \rightarrow \tau_1^m, \tau_2^m \rightarrow \tau_2^m, \ldots, \tau_k^m \rightarrow \tau_k^m.$$

Let $x_{r_1}, \ldots, x_{r_s}$ be the gates of C made true by the assignment v, with $\{r_i\}$ an increasing sequence, so $r_s = n$. Let t be such that $r_t \leq k < r_{t+1}$. Let Λ be the cedent containing the formulas $x_{r_\ell} \rightarrow x_{r_\ell}$ for $\ell > t$. We claim there is a cut-free tree-like proof P_1 of the sequent

$$\Lambda, \Delta, x_{r_1}, \ldots, x_{r_t} \Rightarrow x_{r_s}$$

which has step-length $O(n^2)$ and symbol-length $O(n^3)$. This proof is obtained as follows: First prove

$$\Lambda, \Delta, x_{r_1}, \ldots, x_{r_s} \Rightarrow x_{r_s}$$

from the initial sequent $x_{r_s} \Rightarrow x_{r_s}$ using weakenings. Then derive successively the sequents S_ℓ:

$$\Lambda, \Delta, x_{r_1}, \ldots, x_{r_\ell} \Rightarrow x_{r_s}$$

for $\ell = s-1, \dots, t$. The derivation of $S_{\ell-1}$ from S_ℓ proceeds as follows: (a) if OP_i is $\vee$, use an $\rightarrow$:left inference with one of the sequents

$$x_{j_{r_\ell,u}} \rightarrow x_{r_\ell}, x_{j_{r_\ell,u}} \Rightarrow x_{r_\ell}$$

(with $u \in \{1,2\}$) and then contract the new duplicate formulas in the antecedent; and (b) if OP_i is $\wedge$, do the same thing but with the sequent

$$(x_{j_{r_\ell,1}} \wedge x_{j_{r_\ell,2}}) \rightarrow x_{r_\ell}, x_{j_{r_\ell,1}}, x_{j_{r_\ell,2}} \Rightarrow x_{r_\ell}.$$

Examination of the proof P_1 shows that it has step-length $O(n^2)$ and symbol-length $O(n^3)$, where the extra factor of n allows for lots of exchanges at each stage of the proof.

Now replace every occurrence of variables x_i with $i \leq t$ in P_1 with the formulas τ_i^m. This gives a proof P_2 of the sequent

$$\Lambda, \Delta, T_I^m \Rightarrow x_n.$$

The step-length of P_2 is still $O(n^2)$ and its symbol-length is now $O(m \cdot n^3)$. Now derive the ρ sequents $\Rightarrow \tau_i^m$ for $i \in I$. Combine these with P_2 using $\rightarrow$:left inferences and weakenings to get a proof P_3 of the sequent

$$\Gamma^m, \Lambda, \Delta \Rightarrow x_n.$$

The proof P_3 has step-length $O(\rho \cdot m + n^2)$ and symbol-length $O(\rho \cdot m^2 + m \cdot n^3)$. Finally, the sequent $\Rightarrow \psi$ is easily proved by adding only a small number of inferences to the end of P_3: this increases the lengths of P_3 by at most a constant factor. $\dashv$

The cut-free proofs constructed in the proof of Theorem 20 included initial sequents of the forms $\tau_i^m \Rightarrow \tau_i^m$, so therefore our arguments only work for variations of the sequent calculus which allow arbitrary formulas A in initial sequents $A \Rightarrow A$. If we worked with dag-like sequent calculus proofs, we could get by with proving initial sequents $\tau_i^m \Rightarrow \tau_i^m$ only once for each $i \in I$ and our upper bounds would still hold. However, our proof methods do not work for tree-like, cut-free sequent calculi which allow only atomic formulas in initial sequents.

§**6. The hardness of polynomial calculus.** The polynomial calculus (PC) [4, 13] is based on the idea of converting a CNF formula into an equivalent family of polynomial equations over a field. Let $C = C_1 \wedge C_2 \wedge \dots \wedge C_m$ be a propositional formula over $\{x_1, \dots, x_n\}$, in conjunctive normal form, where each C_i is a clause of size at most three. Each clause C_i is converted into an equation, $\overline{C_i} = 0$ over F such that C is unsatisfiable if and only if $\{\overline{C}_1 = 0, \dots, \overline{C}_m = 0\}$, has no $0/1$ solution. The equations $Q = \{Q_1 = 0, \dots, Q_R = 0\}$ corresponding to C are: $\{\overline{C}_1 = 0, \dots, \overline{C}_m = 0\}$, plus the equations $x^2 - x = 0$ for all variables x. Here is a simple example. Let $C = (b \vee a) \wedge (\neg a \vee b) \wedge (\neg b)$. Then $Q = \{Q_1 = 0, Q_2 = 0, \dots, Q_5 = 0\}$, where $Q_1 = (1-b)(1-a) = 1-a-b+ab$, $Q_2 = (a)(1-b) = a-ab$, $Q_3 = b$, $Q_4 = a^2 - a$, $Q_5 = b^2 - b$.

An *algebraic* refutation for C (over a fixed field F) is an algebraic straight-line program, $S = S_1, \dots, S_l$ such that each S_i is either one of the initial equations (from Q) or is obtained from previous equations by a valid rule, and where the final equation S_l is $1 = 0$. The two rules are as follows. (1) From $g_1(\overline{x}) = 0$ and $g_2(\overline{x}) = 0$, derive $ag_1(\overline{x}) + bg_2(\overline{x}) = 0$, where a, b are constants from F; (2) From

$g(\overline{x}) = 0$, infer $xg(\overline{x}) = 0$ for x a variable. (Thus, a proof is merely an explicit derivation that 1 is in the ideal generated by Q.) In the above example, a refutation is: $S_1 = Q_1$, $S_2 = Q_2$, $S_3 = Q_3$, $S_4 = S_1 + S_2 = 1 - b$, $S_5 = S_4 + S_3 = 1$.

The algebraic proof system is *sound* and *complete*. (See [4] for proofs.) The *algebraic size* of a refutation is the the number of lines, l, in S. The *degree* is defined to be the maximum degree of the intermediate polynomials S_i, after simplifications. This measure has been studied quite a bit, and the name *Polynomial Calculus* (PC) is given to algebraic proofs in this form, where the S_i's are viewed as explicit sums of monomials. The *monomial size* is the total number of monomials in the PC refutation (the sum of the sizes of the S_i's).

The polynomial calculus is not known to be automatizable; however [13] show that constant-degree PC is automatizable with respect to both algebraic and monomial size. We show here that one cannot approximate the minimum PC proof size, to within a linear factor. The proof can also be carried out for other notions of size. The argument essentially mimics the corresponding argument for Resolution.

LEMMA 21. *Let φ be an instance of Circuit MMSA and let ρ equal the cardinality of the minimum satisfying assignment for φ. Γ_φ has a PC refutation with algebraic or monomial size equal to $O(\rho m + n)$.*

PROOF. Recall from the proof of Theorem 5, that Γ_φ has a dag-like resolution refutation of size $O(\rho m + n)$. Furthermore, the width of every intermediate clause in the resolution refutation is at most 3. We will simulate the resolution refutation, line-by-line, by a PC refutation to obtain a size $O(\rho m + n)$ PC refutation. Each clause in the resolution refutation converts into a degree 3 polynomial equation, and hence each equation has constant size. Moreover, the line-by-line simulation also has only a constant factor overhead. ⊣

LEMMA 22. *Let φ, and ρ be as above. Then any PC refutation must have algebraic or monomial size at least ρm.*

PROOF. The proof of this lemma is almost identical to the proof of the corresponding lemma for resolution. ⊣

The above two lemmas imply the following theorem.

THEOREM 23. *If $P \neq NP$, then there is no polynomial time algorithm which can approximate Minimum Size PC Refutation to within $2^{\log^{1-o(1)} n}$ factor.*

§7. Hardness results for long proofs. In the previous sections we proved that it is NP-hard to approximate the minimal propositional proof length to within a $2^{\log^{(1-\varepsilon)} n}$ factor. The tautologies used in the proofs of these results had "short" proofs (or refutations); that is, proofs whose length is polynomial in the size of the formula. However, if $NP \neq coNP$, then for any proof system $\mathcal{S}$, there are tautologies whose shortest $\mathcal{S}$-proof is of super-polynomial length. It is therefore interesting to ask whether better non-approximability results can be achieved when the proof lengths are not bounded, and when the run time of the algorithm is required to be polynomial time in the length of the input formula only.

The following simple intuition implies that in this case, no polynomial time algorithm can guarantee a polynomial time approximation for the shortest refutation of a given unsatisfiable formula, unless $NP \not\subseteq P/poly$[3]:

Given an input formula ψ of length n (an input to SAT), reduce it to a formula $\varphi = \psi \wedge \eta$, such that η is unsatisfiable but its shortest refutation is larger than the refutation of any unsatisfiable formula of length n by a super-polynomial factor. Then ψ is unsatisfiable iff on input φ, a supposed polynomially bounded approximation algorithm returns a number smaller than the size of the shortest refutation of η. This implies a polynomial time circuit for recognizing SAT. To make the above argument formal, we need a few more definitions.

DEFINITION. For a proof system $\mathcal{S}$ and an unsatisfiable formula φ, $\min_{\mathcal{S}}(\varphi)$ is the minimum length of a refutation of φ in $\mathcal{S}$. For an integer n, $MAX_{\mathcal{S}}(n) = \max\{\min_{\mathcal{S}}(\varphi)\}$, where φ ranges over all unsatisfiable formulas of length $\leq n$.

We say that a non-decreasing function f has *super-polynomial growth* if for every polynomial r, $f(n) > r(n)$ for almost all positive integers n. The function f has a *smooth* super-polynomial growth if in addition there is a constant D such that for each n there is $1 < d < D$ such that $f(n^d) > f^d(n)$. [If we write $f(n) = n^{e(n)}$, then the first condition states that $e(n)$ is not bounded from above, and the second condition states that for each n there is m, $n < m < n^D$, such that $e(m) > e(n)$.]

Assume, for simplicity, that $\mathcal{S}$ contains the connective $\wedge$. Formulas ψ and η are said to be *disjoint* if their underlying sets of variables are disjoint.

THEOREM 24. *Assume that $NP \not\subseteq P/poly$, and let $\mathcal{S}$ be a proof system which satisfies*:

1. *For every pair of disjoint formulas ψ and η, where η is unsatisfiable, the following holds*:
   (a) *If ψ is unsatisfiable, then* $\min_{\mathcal{S}}(\psi \wedge \eta) < \min_{\mathcal{S}}(\psi) + r(|\psi| + |\eta|)$ *for some (fixed) polynomial r.*
   (b) *If ψ is satisfiable, than* $\min_{\mathcal{S}}(\psi \wedge \eta) \geq \min_{\mathcal{S}}(\eta)$;
2. *$MAX_{\mathcal{S}}(n)$ has a smooth super-polynomial growth.*

Then for any polynomial q, there is no polynomial time q-approximation algorithm for the minimum length proof in $\mathcal{S}$.

Observe that property 1 above holds trivially for all proof systems mentioned in this paper. Property 2 is known to hold for resolution, since in this case $MAX_{\mathcal{S}}(n) < 3^n$ for all n, and by [5], for each n there is an e, $1 < e < 3$ s.t. $MAX_{\mathcal{S}}(n^e) > 2^{\frac{n^e}{40}}$, thus property 2 holds for $D = 3$ for all large enough n's. We conjecture it to be valid for any known proof system in which the proof lengths are not polynomially bounded.

PROOF. We show that the existence of a polynomial time q-approximation algorithm, AL, for $\mathcal{S}$, implies polynomial time circuits for solving SAT.

Let j be such that $q(n) < n^j$ for almost all n, and let D be the constant guaranteed by the smooth super-polynomial growth of $MAX_{\mathcal{S}}$. Since $MAX_{\mathcal{S}}$ has super-polynomial growth, for all large enough n it holds that $r(n + n^{2jD}) < MAX_{\mathcal{S}}(n)$.

[3]We present the results in terms of finding short refutations of unsatisfiable formulas, but equivalent definitions and results are easily obtained for finding short proofs of tautologies.

Fix an integer n_0 for which this inequality holds. Since the super-polynomial growth of $MAX_{\mathcal{S}}$ is smooth, there is a number d, $2j \leq d \leq 2jD$, such that $[MAX_{\mathcal{S}}(n_0)]^d < MAX_{\mathcal{S}}(m)$, where $m = {n_0}^d$. Let η_m be a formula of size $\leq m$ such that $\min_{\mathcal{S}}(\eta_m) = MAX_{\mathcal{S}}(m)$. An input formula ψ of size n_0 is reduced to $\varphi = \psi \wedge \eta_m$, where the variables of η_m are disjoint from these of ψ (note that φ is unsatisfiable and its size is polynomial in that of ψ). We claim that ψ is unsatisfiable if and only if AL on input φ will output a number $k < MAX_{\mathcal{S}}(m)$. To see this, observe that if ψ is unsatisfiable, then by property (1a) above, $\min_{\mathcal{S}}(\varphi) \leq \min_{\mathcal{S}}(\psi) + r(|\psi| + |\eta_m|) < 2MAX_{\mathcal{S}}(n_0)$. Hence, by the assumption on AL, AL must produce an output $k < (2MAX_{\mathcal{S}}(n_0))^j < MAX_{\mathcal{S}}(m) = \min_{\mathcal{S}}(\eta_m)$. On the other hand, if ψ is satisfiable, then, by property (1b), $\min_{\mathcal{S}}(\varphi) \geq \min_{\mathcal{S}}(\eta_m) = MAX_{\mathcal{S}}(m)$. ⊣

§8. **Acknowledgments.** We are grateful to A.A. Razborov for extremely helpful discussions. We also would like to thank S. Arora for pointing out that Minimum Label Cover can be reduced to Monotone Minimum Satisfying Assignment.

REFERENCES

[1] M. Alehknovich, S. Buss, S. Moran, and T. Pitassi, *Minimum propositional proof length is NP-hard to linearly approximate (extended abstract)*, ***Mathematical foundations of computer science (mfcs′98)***, Lecture Notes in Computer Science #1450, Springer Verlag, 1998, pp. 176–184.

[2] S. Arora, L. Babai, J. Stern, and Z. Sweedyk, *The hardness of approximate optima in lattices, codes, and systems of linear equations*, ***Journal of Computer and System Sciences***, vol. 54 (1997), pp. 317–331, Earlier version in *Proceedings of the 34th Symposium on the Foundations of Computer Science*, 1993, pp.724-733.

[3] S. Arora and C. Lund, *Hardness of approximations*, ***Approximation algorithms for NP-hard problems*** (D. S. Hochbaum, editor), PWS Publishing Co., Boston, 1996.

[4] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák, *Lower bounds on Hilbert's Nullstellensatz and propositional proofs*, ***Proceedings of the London Mathematical Society***, vol. 73 (1996), pp. 1–26.

[5] P. Beame and T. Pitassi, *Simplified and improved resolution lower bounds*, ***37th annual symposium on foundations of computer science***, IEEE Computer Society Press, Los Alamitos, California, 1996, pp. 274–282.

[6] M. Bellare, *Proof checking and approximation: Towards tight results*, ***SIGACT News***, vol. 27 (1996), pp. 2–13, Revised version at `http://www-cse.ucsd.edu/users/mihir`.

[7] M. Bellare, S. Goldwasser, C. Lund, and A. Russell, *Efficient probabalistically checkable proofs and applications to approximation*, ***Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing***, Association for Computing Machinery, 1993, pp. 294–304.

[8] M. Bonet, R. Gavalda, C. Domingo, A. Maciel, and T. Pitassi, *No feasible interpolation or automatization for bounded-depth frege systems*, 1998, Manuscript.

[9] M. L. Bonet, *The Lengths of Propositional Proofs and the Deduction Rule*, ***Ph.D. thesis***, U.C. Berkeley, 1991.

[10] M. L. Bonet, T. Pitassi, and R. Raz, *No feasible interpolation for TC^0-Frege proofs*, ***Proceedings of the 38th annual symposium on foundations of computer science***, IEEE Computer Society, Piscataway, New Jersey, 1997, pp. 264–263.

[11] S. R. Buss, *Some remarks on lengths of propositional proofs*, ***Archive for Mathematical Logic***, vol. 34 (1995), pp. 377–394.

[12] S.R. Buss, *On Gödel's theorems on lengths of proofs. II. Lower bounds for recognizing k symbol provability*, ***Feasible mathematics II*** (P. Clote and J. Remmel, editors), Birkhäuser Boston, Boston, MA, 1995, pp. 57–90.

[13] M. Clegg, J. Edmonds, and R. Impagliazzo, *Using the Groebner basis algorithm to find proofs of unsatisfiability*, ***Proceedings of the twenty-eighth annual acm symposium on the theory of computing***

(New York), Association for Computing Machinery, 1996, pp. 174–183.

[14] S. A. COOK and R. A. RECKHOW, *The relative efficiency of propositional proof systems*, this JOURNAL, vol. 44 (1979), pp. 36–50.

[15] I. DINUR and S. SAFRA, *On the hardness of approximating label cover*, ***Technical Report TR99-015***, ECCC, 1999, `http://www.eccc.uni-trier.de`.

[16] M. H. GOLDWASSER and R. MOTWANI, *Intractability of assembly sequencing: Unit disks in the plane*, ***Proceedings of the Workshop on Algorithms and Data Structures***, Lecture Notes in Computer Science #1272, Springer-Verlag, 1997, pp. 307–320.

[17] ———, *Complexity measures for assembly sequences*, ***International Journal of Computational Geometry & Applications***, vol. 9 (1999), pp. 371–417.

[18] K. IWAMA, *Complexity of finding short resolution proofs*, ***Mathematical foundations of computer science 1997***, Lecture Notes in Computer Science #1295, Springer-Verlag, 1997, pp. 309–318.

[19] K. IWAMA and E. MIYANO, *Intractibility of read-once resolution*, ***Proceedings of the Tenth Annual Conference on Structure in Complexity Theory***, IEEE Computer Society, Los Alamitos, California, 1995, pp. 29–36.

[20] S. KHANNA, M. SUDAN, and L. TREVISAN, *Constraint satisfaction: the approximability of minimization problems*, ***Twelfth annual ieee conference on computational complexity***, IEEE Computer Society, 1997, pp. 282–296.

[21] J. KRAJÍČEK and P. PUDLÁK, *Some consequences of cryptographical conjectures for* S^1_2 *and EF*, ***Logic and computational complexity*** (D. Leivant, editor), Lecture Notes in Computer Science #960, Springer-Verlag, Berlin, 1995, pp. 210–220.

[22] C. LUND and M. YANNAKAKIS, *On the hardness of approximating minimization problems*, ***Journal of the Association for Computing Machinery***, vol. 41 (1994), pp. 960–981.

[23] R. RAZ and S. SAFRA, *A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP*, ***Proceedings of the 29-th Annual ACM Symposium on Theory of Computing***, 1997, pp. 475–484.

[24] R. A. RECKHOW, *On the Lengths of Proofs in the Propositional Calculus*, ***Ph.D. thesis***, Department of Computer Science, University of Toronto, 1976, Technical Report #87.

[25] R. STATMAN, *Complexity of derivations from quantifier-free Horn formulae, mechanical introduction of explicit definitions, and refinement of completeness theorems*, ***Logic colloquium '76*** (R. Gandy and M. Hyland, editors), North-Holland, Amsterdam, 1977, pp. 505–517.

FACULTY OF MECHANICS & MATHEMATICS
MOSCOW STATE UNIVERSITY, RUSSIA
E-mail: michael@mail.dnttm.ru

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF CALIFORNIA, SAN DIEGO
E-mail: sbuss@ucsd.edu

DEPARTMENT OF COMPUTER SCIENCE
TECHNION, ISRAEL INSTITUTE OF TECHNOLOGY
HAIFA, ISRAEL 32000
E-mail: moran@cs.technion.ac.il

COMPUTER SCIENCE DEPARTMENT
UNIVERSITY OF ARIZONA
TUCSON, AZ 85721, USA
E-mail: toni@cs.arizona.edu