Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 121 (2023)

Depth d Frege systems are not automatable unless $\mathsf{P} = \mathsf{NP}$

Theodoros Papamakarios[1]

[1]Department of Computer Science, University of Chicago, Chicago (papamakarios@uchicago.edu)

August 19, 2023

Abstract

We show that for any integer $d > 0$, depth d Frege systems are NP-hard to automate. Namely, given a set S of depth d formulas, it is NP-hard to find a depth d Frege refutation of S in time polynomial in the size of the shortest such a refutation. This extends the result of Atserias and Müller [JACM, 2020] for the non-automatability of resolution — a depth 1 Frege system — to any depth $d > 0$ Frege system.

1 Introduction

Since its inception as child discipline of mathematical logic, computability, and by extension complexity theory, has had the following two questions at its core: First, broadly asked, how hard is it to prove a theorem, and second, knowing that a proof exists, how hard is it to find one. Significantly refining earlier results, most notably [1], Atserias and Müller [2] showed that a version of the latter question, even for a system as weak as resolution, is the same as asking whether $\mathsf{P} = \mathsf{NP}$.

Namely, a proof system σ is *automatable* if there is an algorithm that given a provable formula ϕ, constructs a proof of ϕ in σ in time polynomial in the size of the smallest proof of ϕ in σ. What Aterias and Müller show is that resolution is not automatable unless $\mathsf{P} = \mathsf{NP}$.

Now, it seems plausible that the more complicated the proof systems is, the harder it is to automate it. We show that this is not the case with respect to resolution and depth d Frege systems: For any d, depth d Frege systems are automatable if and only if resolution (a depth 1 Frege system) is automatable. More specifically, we extend the Atserias-Müller result, showing that depth d Frege systems are NP-hard to automate.

The Atserias-Müller result has been extended to cutting planes [11], Res(k) [10], and various algebraic proof systems [7]. Whether it can be extended to bounded depth Frege systems had remained open. It should be noted that the non automatability of bounded depth Frege systems was known under a stronger assumption, namely the hardness of factoring Blum integers [4]. But, this result and ours are incomparable; [4] rules out even the weak automatability of bounded depth Frege

ISSN 1433-8092

systems, which is to say that no system polynomially simulating a depth d Frege system is automatable.

Proof outline

The proof is a reduction from SAT. We want for any positive integer d, given a CNF formula F, to construct a formula G such that if F is satisfiable, then G has small depth d refutations, while if F is unsatisfiable, then G requires large depth d refutations. For $d = 1$, [2] considers the formula $\mathrm{Ref}(F, s)$ expressing that F has a resolution refutation of size s. Then a "relativization" construction is applied to $\mathrm{Ref}(F, s)$ to get the formula $\mathrm{RRef}(F, s)$. To show a lower bound for $\mathrm{RRef}(F, s)$ in the case F is unsatisfiable, it is argued in [2], first that there cannot be resolution refutation of $\mathrm{RRef}(F, s)$ having small index-width, and second, following [6], that if $\mathrm{RRef}(F, s)$ had a small resolution refutation, then $\mathrm{Ref}(F, s')$, where s' is polynomially related to s, would have a resolution refutation of small index-width. Notice that arguing in terms of a variant of width, index-width in this case, is necessary. The same argument could not have worked for width, since resolution is automatable with respect to width.

To extend the above for the case $d > 1$, an idea is to employ the construction of [14] (see also [3]), replacing every variable of $\mathrm{RRef}(F, s)$ with Sipser functions of suitable depth. Following [14, 3], one gets a lower bound by repeated applications of Håstad's switching lemma [12], which reduce a size lower bound to essentially a width lower bound. In our case, we need a reduction in the base case of the argument to a lower bound for the index-width, and trying to apply Håstad's switching lemma for index-width instead of width, one encounters several difficulties. We surpass these difficulties by applying the weaker Furst-Saxe-Sipser switching lemma [8]. This will give a weaker lower bound, only polynomial in our case, which nonetheless is sufficient for the purposes of showing non automatability.

2 Bounded depth Frege systems and automatability

2.1 Basic definitions

We assume that formulas are built from constants 0 and 1, propositional variables and their negations, unbounded conjunctions and unbounded disjunctions. So negations can only appear next to variables. The *depth* of a formula is the maximum nesting of conjunctions and disjunctions in it. Formally,

$$d(0) = d(1) = d(x) = d(\neg x) = 0,$$
$$d(\circ\{F_1, \dots, F_k\}) = 1 + \max_i(F_i),$$

where x is a variable and $\circ$ is either a conjunction $\bigwedge$ or a disjunction $\bigvee$.

Depth 0 formulas that are not constants are called *literals*. We often write literals in the form x^ε, where $x^1 \coloneqq x$ and $x^0 \coloneqq \neg x$. Depth 1 formulas are called *clauses/terms*, clauses being disjunctions and terms conjunctions of literals. Depth 2 formulas that are disjunctions of terms are called *DNF formulas* and depth 2 formulas that are conjunctions of clauses are called *CNF formulas*. DNF formulas each conjunction of which consists of at most k literals are called *k-DNF* formulas; k-CNF formulas are defined similarly. We define $\Sigma_d^{s,k}$ to be the class of all formulas F for which there is a depth d formula G that is semantically equivalent to F, the outermost connective of G is $\bigvee$, and

1. G contains at most s subformulas of depth at least 2;

2. all depth 2 subformulas of G are either k-DNFs or k-CNFs.

Similarly, $\Pi_d^{s,k}$ is defined as the class of all formulas F for which there is a depth d formula G semantically equivalent to F, the outermost connective of which is $\bigwedge$, satisfying the above two conditions.

A *restriction* is an assignment $\rho : V \to \{0,1\}$ of truth values to a set V of variables. For a restriction ρ and a formula F, we denote by $F|_\rho$ the formula resulting by replacing every variable x of F which is in the domain of ρ by $\rho(x)$, and then eliminating constants from $F|_\rho$ using the identities

$$A \vee 0 = A,\ A \vee 1 = 1,\ A \wedge 0 = 0,\ A \wedge 1 = A.$$

We call a restriction that gives a value to all variables a *total assignment*, or simply assignment. For a set S of formulas, we write $S \models F$ if for any total assignment α, $G|_\alpha = 1$ for every $G \in S$ implies $F|_\alpha = 1$. For formulas F and G, we write $F \equiv G$ if F and G are semantically equivalent, i.e. it holds that $F \models G$ and $G \models F$.

2.2 LK proofs

Bounded depth Frege systems are commonly presented as subsystems of sequent calculus (*LK* for short) for propositional logic. We give a Tait-style formulation of LK, where we write cedents as disjunctions. The inference rules of the system are shown in Table 1. There, A and B stand for arbitrary formulas whose top-most connective is $\bigvee$, ϕ stands for an arbitrary propositional formula and Φ stands for a set of propositional formulas. $\overline{\phi}$ is the formula that results from ϕ by exchanging every occurrence of $\bigvee$ with $\bigwedge$ and vice versa, and replacing each literal x^ε with $x^{1-\varepsilon}$.

An LK proof from a set of premises S is a sequence of formulas, called the *lines* of the proof, such that each line either belongs to S or results from earlier lines by one of rules of Table 1. If the last line in a proof is the empty disjunction, then the proof is called a *refutation*. A depth d LK proof is an LK proof each line of which is a formula of depth at most d. The *size* of a proof is the total number of symbols occurring in it.

Of particular importance among depth d LK proofs is the case of depth 1 proofs, called *resolution* proofs. In resolution proofs, lines are clauses, and the only applicable LK rules are the weakening and cut rule, which take the form

$$\frac{C}{C \vee D}, \qquad \frac{C \vee x \qquad D \vee \neg x}{C \vee D}$$

for clauses C and D. In the rightmost rule, also called the resolution rule, we say that $C \vee D$ is the result of *resolving* $C \vee x$ on $D \vee \neg x$ on x.

$$\textit{Axioms}: \quad \frac{}{A \vee \neg A}$$

$$\textit{Weakening}: \quad \frac{A}{A \vee B}$$

$$\bigvee\textit{-introduction}: \quad \frac{A \vee \phi}{A \vee \bigvee \Phi}, \text{ where } \phi \in \Phi$$

$$\bigwedge\textit{-introduction}: \quad \frac{A \vee \phi_1 \qquad \ldots \qquad A \vee \phi_k}{A \vee \bigwedge\{\phi_1, \ldots, \phi_k\}}$$

$$\textit{Cut}: \quad \frac{A \vee \phi \qquad B \vee \overline{\phi}}{A \vee B}$$

Table 1: The rules of LK

We may view a proof as a DAG, by drawing for every line A, edges from the lines A is derived to A. In case a proof DAG is a tree, we refer to the proof as being *tree-like*. The next propositions, due to [14], state that depth d LK proofs and tree-like depth $d+1$ LK proofs can be turned into one another with only a polynomial increase in size.

Proposition 2.1 [14]**.** *A depth d LK proof of a formula F from S of size s can be turned into a depth $d+1$ tree-like LK proof of F from S of size polynomial in s.*

Proposition 2.2 [14, 3]**.** *Let S be a set of formulas of depth at most d and F a formula of depth at most d. A depth $d+1$ tree-like LK proof of a formula F from S of size s can be turned into a depth d LK proof of F from S of size $O(s^2)$.*

2.3 Semantic proofs, variable width and decision trees

A semantic depth d (Frege) proof from a set of formulas S is a sequence of depth d formulas $F_1, \ldots, F_t$ such that for every i, either $F_i \in S$ or there are $j, k < i$ such that $F_j, F_k \models F_i$. Notice that if S consists of depth $d-1$ formulas, then there is a trivial depth d proof of any valid consequence of S, as $\bigwedge S$ can be derived in $|S|-1$ steps. Thus under this formulation, depth d proofs from S are interesting only if S contains depth d formulas not in $\Pi_d^{s,k}$ for any s and k, and indeed, our results pertain to such proofs.

The definitions of lines, size of a proof, refutation, tree-like proofs, apply to semantic proofs as well. The *variable width* of a proof is the maximum number of variables among the lines of the proof.

Unlike size, variable width is an inherently semantic notion. In particular, it is independent of depth: any depth d proof of variable width w can be transformed into a depth 1 proof of (variable) width $O(w)$. In fact, something stronger can be said. A *decision tree* is a binary tree the internal nodes of which are labelled with variables, and the edges with values 0 or 1. Nodes query variables and the edges going from a node to its children are labelled, one with the value 0 and the other with 1, giving an answer to that query. No variable is repeated in a branch so that branches correspond to restrictions, and each branch has a value, 0 or 1, associated with it, so that the decision tree represents a Boolean function. We denote the set of branches of $\mathbf{T}$ having the value v by $\mathrm{Br}_v(\mathbf{T})$. Specifically, we say that a decision tree $\mathbf{T}$ *represents a formula* F if for every branch π of $\mathbf{T}$ with value v, $F|_\pi \equiv v$. The *height* of a decision tree is the length of its longest branch. Notice that if a formula F is represented by a decision tree of height h, then $F \in \Sigma_2^{1,h} \cap \Pi_2^{1,h}$. We write $h(F)$ for the minimum height of a decision tree representing F. The following lemma is shown in [17] for a specific type of depth 2 proofs, but holds for proofs of arbitrary depth, or for that matter, arbitrary sound proofs.

Lemma 2.3. *Let S be a set of clauses each containing at most h literals. If there is a semantic refutation of S each line of which is represented by a decision tree of height at most h, then there is a resolution refutation of S of width at most $3h$.*

Proof. Let $F_1, \ldots, F_t$ be a semantic refutation of S and let $\mathbf{T}_i$ be a decision tree of height at most h representing F_i. We assume that $\mathbf{T}_t$ has a single node having the value 0. For a restriction π, let C_π be the minimal clause falsified by π. We will show that for every i, for every branch $\pi \in \mathrm{Br}_0(\mathbf{T}_i)$, we can derive C_π via a resolution proof of width at most $3h$. Notice that C_π for $\pi \in \mathrm{Br}_0(\mathbf{T}_t)$ is the empty clause, so this construction will give a refutation.

If F_i is a clause C in S, then every $\pi \in \mathrm{Br}_0(\mathbf{T}_i)$ must make every literal in C false, hence C_π is a weakening of C. Assume now that F_i is derived from F_j and F_k and we have derived all clauses C_π for $\pi \in \mathrm{Br}_0(\mathbf{T}_j) \cup \mathrm{Br}_0(\mathbf{T}_k)$. Let $\sigma \in \mathrm{Br}_0(\mathbf{T}_i)$, and let $\mathbf{T}$ be the tree resulting by appending a copy of $\mathbf{T}_k$ at the end of every branch $\pi \in \mathrm{Br}_1(\mathbf{T}_j)$ of $\mathbf{T}_j$. We will use $\mathbf{T}$ to extract a resolution proof of C_σ. More specifically, for every node u of $\mathbf{T}$ such that the path π_u from the root of $\mathbf{T}$ to u corresponds to a restriction that is consistent with σ, we will derive $C_\sigma \vee C_{\pi_v}$. When we reach the root of $\mathbf{T}$ we will have derived C_σ. If u is a leaf of $\mathbf{T}$, then we claim that C_{π_u} is a weakening of some clause C_π for $\pi \in \mathrm{Br}_0(\mathbf{T}_j) \cup \mathrm{Br}_0(\mathbf{T}_k)$. To see this, let $\pi_u = \pi_j \cup \pi_k$, where π_j is the part of π_u that belongs to $\mathbf{T}_j$ and π_k the part that belongs to $\mathbf{T}_k$. Since $F_j, F_k \models F_i$ and π_u is consistent with σ, it cannot be the case that both $\pi_j \in \mathrm{Br}_1(\mathbf{T}_j)$ and $\pi_k \in \mathrm{Br}_1(\mathbf{T}_k)$, otherwise a total assignment extending both π_u and σ would make F_j and F_k true, but F_i false. Suppose now that u is not a leaf of $\mathbf{T}$ and suppose that v and w are its children. Then either π_v and π_w are both consistent with σ, in which case $C_\sigma \vee C_{\pi_u}$ can be derived by resolving $C_\sigma \vee C_{\pi_v}$ and $C_\sigma \vee C_{\pi_w}$ on the variable labelling u, or one of the children, say v, will be consistent with σ and thus $C_\sigma \vee C_{\pi_u}$ will be identical to $C_\sigma \vee C_{\pi_v}$. $\square$

2.4 Automatability and the main result

A proof system σ is called *automatable* [5] if there is an algorithm that given a set of formulas S and a formula ϕ provable from S, outputs a σ-proof of ϕ from S in time polynomial $r + s$, where r is the total size of S and s the size of the shortest σ-proof of ϕ from S.

The main theorem of this paper is the fact that approximating the minimum size of a depth d Frege refutation is NP hard:

Theorem 2.4. *For every integer $d > 0$, there is a polynomial-time computable function which on input a CNF formula F with n variables and m clauses and integers $s, N > 0$ represented in unary, returns a CNF formula $G_d(F; s, N)$ such that*

1. *if F is satisfiable, then there is a depth d LK refutation of $G_d(F; s, N)$ of size*

$$O\left(\left(N^{d+3}s^2n(m + s^2n^3\right)^2\right);$$

2. *if F is not satisfiable, N is an increasing function of n and s is a polynomial in n, every semantic depth d refutation of $G_d(F; s, N)$ must have size at least*

$$N^{\frac{1}{3}\left(\frac{\log s}{\log n}-2\right)^{\frac{1}{d-1}}}$$

for large enough n.

The NP hardness of automating depth d Frege systems follows from Theorem 2.4 by setting $s := n^{(3h)^{d-1}+2}$ and $N := s$ for a large enough constant h (see Theorem 6.1).

We describe the reduction, constructing the formula $G_d(F; s, N)$ from F in Section 3. In Section 4, we show the upper bound of Theorem 2.4, and in Section 5 we show the lower bound. It is important to note that both bounds hold for semantic depth d refutations. The reason we formulate the upper bound in terms of LK refutations is twofold. First, we are able to apply Proposition 2.2; we contend it is much cleaner to first give a depth $d+1$ tree-like LK refutation of our formulas and then convert it to a depth d refutation, rather than directly giving a depth d refutation. Secondly, the notion of automatability is neither monotone nor anti-monotone. Hence it is clear from Theorem 2.4 that the non automatability result applies to any version intermediate between depth d LK and depth d semantic systems.

3 The formulas Ref

Let F be a CNF formula with n variables and m clauses. The key ingredient in the non automatability result of [2] is expressing by a set of clauses $\mathrm{Ref}(F, s)$ the statement that there is a resolution refutation $D_1, \ldots, D_s$ of length s from the clauses of F.

The variables of $\mathrm{Ref}(F, s)$ are $D[u, i, b]$, $V[u, i]$, $I[u, j]$, $L[u, v]$ and $R[u, v]$, where $u, v \in [s]$, $i \in [n]$, $j \in [m]$ and $b \in \{0, 1\}$. The meaning of $D[u, i, b]$ is that x_i^b appears in D_u. The meaning of $V[u, i]$ is that D_u is derived as a weakening of the resolvent of two previous clauses on x_i, and the meaning of $I[u, j]$ is that D_u is a weakening of the j-th clause of F. The meaning of $L[u, v]$ is that the left clause (i.e. that which contains $\neg x_i$) from which D_u was derived is D_v, and the meaning of $R[u, w]$ is that the right clause (i.e. that which contains x_i) from which D_u was derived is D_w. We will also use the variables $V[u, 0]$ and $I[u, 0]$ to indicate whether D_u is derived from previous clauses or from an initial clause of F: in the former case, $I[u,0]$ will be true and $V[u,0]$ false, and in the latter $V[u,0]$ will be false and $I[u,0]$ true. The clauses of $\mathrm{Ref}(F,s)$ encode the following conditions: For each $u,v \in [s]$, $i,i' \in [n]$, $j \in [m]$ and $b \in \{0,1\}$,

$$\exists! k\, V[u,k] \ \ \&\ \ \exists! k\, I[u,k] \ \ \&\ \ \exists! k\, L[u,k] \ \ \&\ \ \exists! k\, R[u,k]; \tag{3.1}$$
$$V[u,0] \iff \neg I[u,0]; \tag{3.2}$$
$$\neg L[u,v] \text{ for } v \geq u \ \ \&\ \ \neg R[u,v] \text{ for } v \geq u; \tag{3.3}$$
$$V[u,i] \ \ \&\ \ L[u,v] \implies D[v,i,0]; \tag{3.4}$$
$$V[u,i] \ \ \&\ \ R[u,v] \implies D[v,i,1]; \tag{3.5}$$
$$V[u,i] \ \ \&\ \ L[u,v] \ \ \&\ \ D[v,i',b] \ \ \&\ \ i \neq i' \implies D[u,i',b]; \tag{3.6}$$
$$V[u,i] \ \ \&\ \ R[u,v] \ \ \&\ \ D[v,i',b] \ \ \&\ \ i \neq i' \implies D[u,i',b]; \tag{3.7}$$
$$I[u,j] \ \ \&\ \ x_i^b \text{ appears in } C_j \implies D[u,i,b]; \tag{3.8}$$
$$\neg D[u,i,0] \vee \neg D[u,i,1]; \tag{3.9}$$
$$\neg D[s,i,b]. \tag{3.10}$$

It was shown, subsequent to [2] that $\mathrm{Ref}(F,s)$ is hard for resolution whenever F is unsatisfiable [9]. In [2], a variation, $\mathrm{RRef}(F,s)$, is used. $\mathrm{RRef}(F,s)$ expresses the fact that there is a resolution refutation $D_1,\ldots,D_s$ or one contained in $D_1,\ldots,D_s$, from the clauses of F. $\mathrm{RRef}(F,s)$ has the same variables as $\mathrm{Ref}(F,s)$ plus a new variable $P[u]$ indicating which of the indices $1,\ldots,s$ are active, i.e. are part of the refutation. The clauses of $\mathrm{RRef}(F,s)$ express the following conditions, which are those of $\mathrm{Ref}(F,s)$ conditioned on the fact that $P[u]$ is true, in addition to three new ones requiring $P[s]$ to be true, and $P[v]$ to be true whenever $P[u]$ and $L[u,v]$ or $R[u,v]$ are true:

$$P[u] \implies \exists! k\, V[u,k] \ \ \&\ \ \exists! k\, I[u,k] \ \ \&\ \ \exists! k\, L[u,k] \ \ \&\ \ \exists! k\, R[u,k]; \tag{3.11}$$
$$P[u] \implies (V[u,0] \iff \neg I[u,0])\,; \tag{3.12}$$
$$P[u] \implies \neg L[u,v] \text{ for } v \geq u \ \ \&\ \ \neg R[u,v] \text{ for } v \geq u; \tag{3.13}$$
$$P[u] \implies (V[u,i] \ \ \&\ \ L[u,v] \implies D[v,i,0])\,; \tag{3.14}$$
$$P[u] \implies (V[u,i] \ \ \&\ \ R[u,v] \implies D[v,i,1])\,; \tag{3.15}$$
$$P[u] \implies \big(V[u,i] \ \ \&\ \ L[u,v] \ \ \&\ \ D[v,i',b] \ \ \&\ \ i \neq i' \implies D[u,i',b]\big)\,; \tag{3.16}$$
$$P[u] \implies \big(V[u,i] \ \ \&\ \ R[u,v] \ \ \&\ \ D[v,i',b] \ \ \&\ \ i \neq i' \implies D[u,i',b]\big)\,; \tag{3.17}$$
$$P[u] \implies \Big(I[u,j] \ \ \&\ \ x_i^b \text{ appears in } C_j \implies D[u,i,b]\Big)\,; \tag{3.18}$$
$$P[u] \implies (\neg D[u,i,0] \vee \neg D[u,i,1])\,; \tag{3.19}$$
$$P[s] \ \ \&\ \ \neg D[s,i,b]; \tag{3.20}$$
$$(P[u] \ \ \&\ \ L[u,v] \implies P[v]) \ \ \&\ \ (P[u] \ \ \&\ \ R[u,v] \implies P[v])\,. \tag{3.21}$$

Notice that giving truth values to the $P[u]$ variables (where $P[s]=1$) reduces $\mathrm{RRef}(F,s)$ to $\mathrm{Ref}(F,s')$ where s' is the number of indices u for which $P[u]=1$.

For an integer $k \geq 1$, we define $\mathrm{R}^k\mathrm{Ref}(F,s)$ as the formula resulting from substituting each variable $P[u]$ in $\mathrm{RRef}(F,s)$ with the conjunction $\bigwedge_{i=1}^{k} P_i[u]$ for new variables $P_1[u],\ldots,P_k[u]$. Note that $\mathrm{RRef}(F,s) = \mathrm{R}^1\mathrm{Ref}(F,s)$.

Now, let $d, N \geq 1$ be integers, and let x be a propositional variable. We associate with x $N^{d-1}\lceil\sqrt{N}/2\rceil$ new variables $x_{i_1,\ldots,i_d}$, where $i_1, \ldots, i_{d-1} \in [N]$ and $i_d \in [\lceil\sqrt{N}/2\rceil]$. The fact that we make i_d range over $[\lceil\sqrt{N}/2\rceil]$ instead of $[N]$ will be important later (specifically in Lemma 5.2). The depth d Sipser functions for x are defined by

$$S^{\wedge}_{d,N}(x) \stackrel{\text{def}}{=} \bigwedge_{i_1=1}^{N} \bigvee_{i_2=1}^{N} \cdots \bigwedge_{i_d=1}^{\lceil\sqrt{N}/2\rceil} x_{i_1,\ldots,i_d},$$

$$S^{\vee}_{d,N}(x) \stackrel{\text{def}}{=} \bigvee_{i_1=1}^{N} \bigwedge_{i_2=1}^{N} \cdots \bigvee_{i_d=1}^{\lceil\sqrt{N}/2\rceil} x_{i_1,\ldots,i_d}$$

if d is odd, and

$$S^{\wedge}_{d,N}(x) \stackrel{\text{def}}{=} \bigwedge_{i_1=1}^{N} \bigvee_{i_2=1}^{N} \cdots \bigvee_{i_d=1}^{\lceil\sqrt{N}/2\rceil} x_{i_1,\ldots,i_d},$$

$$S^{\vee}_{d,N}(x) \stackrel{\text{def}}{=} \bigvee_{i_1=1}^{N} \bigwedge_{i_2=1}^{N} \cdots \bigwedge_{i_d=1}^{\lceil\sqrt{N}/2\rceil} x_{i_1,\ldots,i_d}$$

if d is even.

We define $\mathrm{RRef}_{d,N}(F, s)$ to be the result of substituting every variable of the form $P[u]$ in $\mathrm{RRef}(F, s)$ with $S^{\wedge}_{d,N}(P[u])$ and every other variable x with $S^{\vee}_{d,N}(x)$. Notice that $\mathrm{RRef}_{d,N}(F, s)$ is a set of depth $d + 1$ formulas. But, as we want to prove statements about whether $\mathrm{RRef}_{d,N}(F, s)$ has or does not have small depth d refutations, we must write it as a set of depth d formulas. We may do that with only a polynomial increase in size, as the only clauses of non constant size of $\mathrm{RRef}(F, s)$ are those of the form $\neg P[u] \vee \bigvee_i X[u, i]$ corresponding to conditions (3.11), and these clauses will have depth d after the substitution taking us from $\mathrm{RRef}(F, s)$ to $\mathrm{RRef}_{d,N}(F, s)$. Note that the conversion from $\mathrm{RRef}_{d,N}(F, s)$ written as a set of depth d formulas to its equivalent set of depth $d + 1$ formulas can be carried in tree-like depth $d + 1$ LK in linear time. In particular, a tree-like depth $d + 1$ LK refutation of the latter set can be turned into a tree-like depth $d+1$ LK refutation of the former set, increasing the size by at most a factor of N^3.

4 Upper bounds

We show in this section that if F is satisfiable, then $\mathrm{RRef}_{d,N}(F, s)$ has small depth d refutations:

Proposition 4.1. *If F is a satisfiable CNF formula with n variables and m clauses, then there is a depth d LK refutation of $\mathrm{RRef}_{d,N}(F, s)$ of size*

$$S = O\left(\left(N^{d+3}s^2n(m + s^2n^3)\right)^2\right).$$

In particular, if $m = O(s^2n^3)$, then $S = O(N^{2(d+3)}(sn)^8)$.

Proof. We start with a small depth 2 LK tree-like refutation of $\mathrm{RRef}(F, s)$. This refutation will be such that after the substitution with Sipser functions, we get a depth $d+1$ tree-like refutation of $\mathrm{RRef}_{d,N}(F, s)$, which in turn we can convert to a depth d DAG-like refutation of $\mathrm{RRef}_{d,N}(F, s)$ by Proposition 2.2.

We write, for better readability, $A_1, \dots, A_k \to B_1, \dots, B_\ell$ instead of $\overline{A_1} \vee \dots \vee \overline{A_k} \vee B_1 \vee \dots \vee B_\ell$.

Let α be an assignment that satisfies every clause of F. We set

$$T(u) \coloneqq P[u] \to \bigvee_{i=1}^{n} D[u, i, \alpha(x_i)].$$

What $T(u)$ says is that if $P[u]$ is true, then α satisfies the u-th clause in the refutation $\mathrm{Ref}(F, s)$ describes.

Our refutation of $\mathrm{RRef}(F, s)$ consists of $s-1$ stages, starting with stage 0. In the u-th stage, $T(1), \dots, T(s-u) \to 0$ will have been derived. Then we can use this formula, along with a derivation of $T(1), \dots, T(s-u-1) \to T(s-u)$, to derive $T(1), \dots, T(s-u-1) \to 0$. In the $s-1$-th stage, $T(1) \to 0$ will have been derived, at which point we can reach a contradiction by deriving $T(1)$.

A derivation of $T(1), \dots, T(v-1) \to T(v)$ is sketched in Figure 1. The formulas

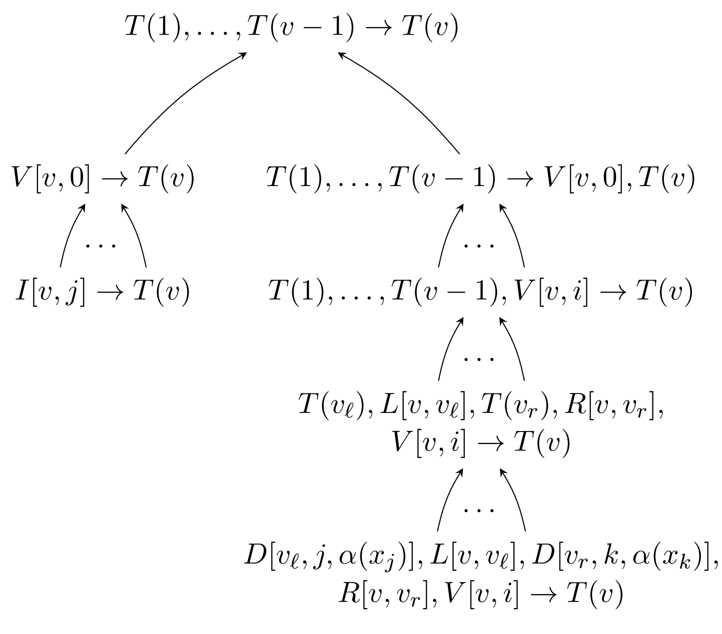

Figure 1: A sketch of a derivation of $T(1), \dots, T(v-1) \to T(v)$

$I[v, j] \to T(v)$ for $j \in [m]$ can be immediately derived from the clauses $P[u] \wedge I[v, j] \to D[v, i, \alpha(x_i)]$, which are clauses corresponding to condition (3.18), as the fact that α satisfies the clause C_j means that $x_i^{\alpha(x_i)}$ must belong to C_j for some i. These formulas can be in turn used along with the clauses (3.11) for $I[v, k]$ and (3.12) to derive $V[v, 0] \to T(v)$. Now deriving

$$T(1), \dots, T(v-1) \to V[v, 0], T(v) \tag{4.1}$$

will allow us to derive $T(1), \ldots, T(v-1) \to T(v)$ by cutting on $V[v, 0]$. We can derive (4.1) from the formulas

$$T(1), \ldots, T(v-1), V[v, i] \to T(v) \tag{4.2}$$

for $i \in [n]$ using the clauses (3.11) for $V[v, i]$. The formulas (4.2) can be in turn derived from the formulas

$$T(v_\ell), L[v, v_\ell], T(v_r), R[u, v_r], V[v, i] \to T(v) \tag{4.3}$$

for $v_\ell, v_r \in [s]$ using the clauses (3.11) for $L[v, k]$ and $R[v, k]$, (3.12) and (3.13). Finally, (4.3) can be derived from the formulas

$$D[v_\ell, j, \alpha(x_j)], L[v, v_\ell], D[v_r, x_k, \alpha(x_k)], R[u, v_r], V[v, i] \to T(v), \tag{4.4}$$

for $j, k \in [n]$, which can be derived directly from the clauses (3.21) and either (3.14), (3.15) and (3.19) or (3.16) and (3.17) depending on whether $i = j = k$ or not.

We can see that the derivations of $T(1), \overline{T(s)}$ and $T(1), \ldots, T(v-1) \to T(v)$ take at most $O(m+s^2n^3)$ steps, hence the overall refutation has size $O\left(s^2n(m + s^2n^3)\right)$.

Now, notice that after substituting every variable $P[u]$ in it with $S^{\wedge}_{d,N}(P[u])$ and every other variable x with $S^{\vee}_{d,N}(x)$, $T(v)$ becomes a depth d formula. Hence we see that after the substitution, the refutation described above becomes a depth $d+1$ tree-like LK refutation of $\mathrm{RRef}_{d,N}(F, s)$. We can then get a depth d refutation of $\mathrm{RRef}_{d,N}(F, s)$ of the required size by applying Proposition 2.2. $\square$

5 Lower bounds

Lower bounds for depth d Frege systems for $d > 1$, typically follow the following strategy:

1. We first show that the formulas we are trying to refute are robust; namely, after applying a restriction selected at random to them, then with high probability they cannot be refuted with proofs whose lines are, in a certain sense, simple.

2. Then we show, through the use of a switching lemma, that applying such a restriction to a short proof will result with high probability in a proof with simple lines.

Here we start with $\mathrm{RRef}_{d,N}(F, s)$, which after applying the restrictions will collapse to $\mathrm{Ref}(F, s')$, where s' is polynomially related to s. For the part of the overall strategy showing that there cannot be refutations with simple lines, we take, as in [2], simple to mean of small *index-width*. We say that a variable of the form $D[u, i, b]$, $V[u, i]$, $I[u, j]$, $L[u, v]$ or $R[u, v]$ *mentions* the index u. The index-width of a clause in the variables of $\mathrm{Ref}(F, s)$ is defined as the number of indices mentioned by its variables, and the index-width of a resolution refutation of $\mathrm{Ref}(F, s)$ is the maximum index-width over its clauses. We have:

Theorem 5.1 [2]**.** *For all integers $n, s > 0$ with $s \leq 2^n$, and every unsatisfiable CNF F with n variables, every resolution refutation of $\mathrm{Ref}(F, s)$ has index-width at least $s/6n$.*

5.1 The robustness of $\mathbf{RRef}_{d,N}(F,s)$

We create a distribution on restrictions to the variables of $\mathrm{RRef}_{d,N}(F,s)$ as follows. Suppose d is odd (if d were even, we would exchange the roles of 0 and 1 in the following construction). For each $S^{\wedge}_{d,N}(x)$ formula in $\mathrm{RRef}_{d,N}(F,s)$, look at its bottom-most N^{d-1} $\bigwedge$ connectives. For each such connective, we decide to "preserve" it with probability $1/\sqrt{N}$, and not to preserve it with probability $1-1/\sqrt{N}$. For each of the preserved connectives, we leave its first variable unset and set the rest to 1. For each variable in the unpreserved connectives, we set it to 0 or 1 with probability 1/2 for each choice. The variables of $S^{\vee}_{d,N}(x)$ are set in the same way, except that the set variables of the preserved $\bigvee$ connectives are set to 0 instead of 1.

Under such restrictions, Sipser functions do not simplify much. For formulas F and G, in which each variable appears only once, we say that F *contains* G if we can get G from F by deleting some of its literals and/or renaming some of its variables.

Lemma 5.2. *For any $d \geq 2$, the probability that $S^{\nu}_{d,N}(x)|_\rho$, where $\nu \in \{\wedge, \vee\}$, does not contain $S^{\nu}_{d-1,N}(x)$ is at most $2^{-\Omega(\sqrt{N})}$.*

Proof. We show the lemma for $S^{\wedge}_{d,N}(x)$ and d odd. If $S^{\wedge}_{d,N}(x)|_\rho$ does not contain $S^{\wedge}_{d-1,N}$, then either one of its bottom-most $\bigwedge$ connectives takes the value 1, or in one of its depth 2 subformulas, less than $\sqrt{N}/2$ $\bigwedge$ connectives are preserved. The probability that a bottom-most $\bigwedge$ connective takes the value 1 is at most $2^{-\sqrt{N}/2}$ and the probability that this happens for at least one of the N^{d-1} bottom-most $\bigwedge$ connectives is at most

$$N^{d-1}2^{-\sqrt{N}/2} \leq 2^{-\Omega(\sqrt{N})}.$$

Now fix a depth 2 subformula A of $S^{\wedge}_{d,N}(x)$. The expected number of preserved $\bigwedge$ connectives in $A|_\rho$ is $N/\sqrt{N} = \sqrt{N}$, and by the Chernoff bound, the probability that there are less than $\sqrt{N}/2$ preserved $\bigwedge$ connectives is at most $2^{-\Omega(\sqrt{N})}$. The probability that at least one of the N^{d-2} depth 2 subformulas of $S^{\wedge}_{d,N}(x)$ has less than $\sqrt{N}/2$ preserved connectives is thus at most

$$N^{d-2}2^{-\Omega(\sqrt{N})} \leq 2^{-\Omega(\sqrt{N})}.$$

We conclude that the probability that $S^{\wedge}_{d,N}|_\rho$ does not contain $S^{\wedge}_{d-1,N}$ is at most $2^{-\Omega(\sqrt{N})}$. □

5.2 The Furst-Saxe-Sipser switching lemma

Switching lemmas provide conditions under which a k-DNF formula "switches" to a ℓ-CNF formula after applying a restriction created at random. We will use the switching lemma of [8] and a variation tailored for $\mathrm{R}^k\mathrm{Ref}(F,s)$ due to [10].

Let G be a k-DNF formula in variables X. Let $X_1,\ldots,X_r$ be a partition of X into r blocks, and let $\nu \in \{0,1\}$. Consider the following distribution over restrictions on X: For each block X_i, we decide to "preserve" X_i with probability p, and not to preserve it with probability $1-p$. For each preserved block, we leave one of its variables, say the first in the block, unset, and set all others to ν. For each unpreserved block, we set each of its variables to 0 or 1 with probability 1/2 for each value.

We can extract the following lemma from [8, 18]. The lemma is implicit in [8, 18] with parameters obscured under a big O notation. We present it here in a more general, improved form, with explicit parameters, using decision trees along the lines of [17]. ln in what follows denotes the natural logarithm; we preserve the notation log for the base 2 logarithm.

Lemma 5.3 (see [8, 18])**.** *If $phk2^k \ln N = o(N^{-\varepsilon})$ for some $\varepsilon \in (0,1)$, then*

$$P[h(G|_\rho) > kh] \le o(N^{-\varepsilon h})\frac{2^{kh}-1}{2^h-1}.$$

Proof. The proof is by induction on k. If $k = 0$, G is a constant and can be represented by a decision tree of height 0. Suppose $k > 0$. We distinguish between two cases, G being wide and G being narrow. We call G wide if there are at least $h2^k \ln N$ terms in it such that no two of them contain variables from the same block. G is narrow iff it is not wide. If G is wide, then

$$\begin{aligned} P[h(G|_\rho) > kh] \le P[G|_\rho \ne 1] &\le \left(1 - \left(\frac{1-p}{2}\right)^k\right)^{h2^k \ln N} \\ &\le e^{-(1-p)^k h \ln N} = o(N^{-\varepsilon h}). \end{aligned}$$

If G is narrow, then take a maximal set of terms such that no two of them contain variables from the same block, and let H be the set of blocks that contain a variable occurring in some term of this set. H contains at most $hk2^k \ln N$ blocks and every term of G contains some variable (or its negation) from some block in H. The probability of the event A that ρ preserves more than h blocks in H is

$$P[A] \le \binom{hk2^k \ln N}{h} p^h \le (hk2^k \ln N)^h p^h = o(N^{-\varepsilon h}).$$

Now, let π be a restriction that sets the variables of all blocks in H, and let A_π be the event that π is consistent with ρ and $h((G|_\rho)|_\pi) > (k-1)h$. Notice that $G|_\pi$ is a $(k-1)$-DNF, so by the induction hypothesis,

$$P[A_\pi] \le P[h((G|_\pi)|_\rho) > (k-1)h] \le o(N^{-\varepsilon h})\frac{2^{(k-1)h}-1}{2^h-1}.$$

We get

$$\begin{aligned} P\left[A \cup \bigcup_\pi A_\pi\right] &\le o(N^{-\varepsilon h}) + o(N^{-\varepsilon h}) 2^h \frac{2^{(k-1)h}-1}{2^h-1} \\ &= o(N^{-\varepsilon h})\frac{2^{kh}-1}{2^h-1}. \end{aligned}$$

In the event

$$\left(A \cup \bigcup_\pi A_\pi\right)^c,$$

i.e. the event that ρ preserves at most h blocks in H and for all restrictions π consistent with ρ, $h((G|_\rho)|_\pi) \leq (k-1)h$, we can construct a decision tree of height at most kh representing $G|_\rho$ as follows: We query all variables belonging to some block in H left unset by ρ (since ρ preserves at most h blocks in H, there are at most h of them), and at each branch π of the resulting tree, we append a decision tree of minimum height representing $(G|_\rho)|_\pi$. $\square$

We create a distribution on restrictions on the variables of $\mathrm{R}^\ell\mathrm{Ref}(F,s)$ as follows: For every index u and every $i \in [\ell]$, we set $P_i[u]$ to 0 or 1, with probability 1/2 for each value. Let U be the set of indices such that $P_i[u] = 1$ for all $i \in [\ell]$. For each variable x of $\mathrm{R}^\ell\mathrm{Ref}(F,s)$ not of the form $P_i[u]$ mentioning an index in U, we set x to 0 or 1, with probability 1/2 for each value.

For a decision tree $\mathbf{T}$ querying variables of $\mathrm{Ref}(F,s)$, we define the *index-height* of $\mathbf{T}$ as the maximum number of indices mentioned by variables over all branches that do not falsify axioms of $\mathrm{Ref}(F,s)$. For a formula G, We denote by $\hbar(G)$ the minimum index height of a decision tree representing G.

The following lemma is from [10]. We give a proof because in [10] the lemma is stated not for $\mathrm{R}^\ell\mathrm{Ref}(F,s)$ but a variation, plus we view the following proof to be simpler.

Lemma 5.4 [10]**.** *Let F be a CNF formula in n variables, k and ℓ integers with $0 < k \leq \ell$, and G a k-DNF formula over the variables of $R^\ell Ref(F,s)$. Then for large enough n,*

$$P[\hbar(G|_\rho) > h] \leq 2^{-\frac{h}{n^{k-1}}\gamma(k)},$$

where $\gamma(0) = 1$, $\gamma(i) = (\log e)(i4^{i+1})^{-1}\gamma(i-1)$.

Proof. Let $h_i \coloneqq h\gamma(i-1)/(4n^{i-1})$. We will show, by induction on k, that for every k and ℓ with $k \leq \ell$, for every k-DNF formula G over the variables of $\mathrm{R}^\ell\mathrm{Ref}(F,s)$,

$$P\left[\hbar(G|_\rho) > \sum_{i=1}^{k} h_i\right] \leq 2^{-\frac{h}{n^{k-1}}\gamma(k)}$$

for large enough n.

If $k = 0$, F is a constant and can be represented by a decision tree of height 0. Suppose $k > 0$. We call G wide if there are at least h_k/k terms in G over disjoint sets of indices, and call G narrow otherwise. Suppose G is wide. A literal in a term t of G is satisfied with probability at least 1/4: Literals on a variable $P_i[u]$ are satisfied with probability 1/2. For any other literal x^ϵ of t mentioning the index u, since $k \leq \ell$, there must be a variable $P_i[u]$ not in t, which is made 0 with probability 1/2, in which case x^ϵ will be satisfied with probability 1/2. Hence

$$\begin{aligned} P[\hbar(G|_\rho) > h] &\leq P[G|_\rho \neq 1] \leq (1-4^{-k})^{\frac{h\gamma(k-1)}{4kn^{k-1}}} \\ &\leq 2^{-\frac{h}{n^{k-1}}(\log e)(k4^{k+1})^{-1}\gamma(k-1)} \\ &= 2^{-\frac{h}{n^{k-1}}\gamma(k)}. \end{aligned}$$

Suppose now that G is narrow. Take a maximal set of terms over disjoint sets of indices, and let H be the set of indices that are mentioned by the terms of this set. Notice that $|H| \leq h_k$ and that every term of G contains some variable (or its negation) that mentions an index in H. Let π be a restriction that

1. sets all variables mentioning an index in H and leaves all other variables unset, and
2. does not falsify any axioms of $\mathrm{R}^\ell Ref(F, s)$.

The second condition means in particular that if U is the set of indices u for which π sets $P_i[u]$ to 1 for all i, then for all $u \in U$, there will be exactly one v such that $L[u, v]$ is true, exactly one v such that $R[u, v]$ is true, exactly one i such that $V[u, i]$ is true, and exactly one j such that $I[u, j]$ is true, making the total number of such π's to be at most

$$S^{|U|}2^{(|H|-|U|)n_0}$$

where $S \coloneqq s^2(n+1)(m+1)2^{2n}$ and n_0 is the number of variables of $\mathrm{R}^\ell\mathrm{Ref}(F, s)$ mentioning a fixed index u.

Let A_π be the event that π is consistent with ρ and $\hbar((G|_\rho)|_\pi) > \sum_{i=i}^{k-1} h_i$. We have that

$$\begin{aligned}
P[A_\pi] &= P\left[\hbar(G|_\rho)|_\pi > \sum_{i=i}^{k-1} h_i \mid \rho \text{ con. with } \pi\right] P[\rho \text{ con. with } \pi] \\
&= P\left[\hbar(G|_\pi)|_\rho > \sum_{i=i}^{k-1} h_i\right] P[\rho \text{ con. with } \pi] \\
&\leq 2^{-\frac{h}{n^{k-2}}\gamma(k-1)}2^{-\ell|H|}2^{-(|H|-|U|)n_0}.
\end{aligned}$$

Hence, we get

$$\begin{aligned}
P\left[\bigcup_\pi A_\pi\right] &\leq \sum_\pi P[A_\pi] \\
&\leq \sum_{U \subseteq H} S^{|U|}2^{(|H|-|U|)n_0}2^{-\frac{h}{n^{k-2}}\gamma(k-1)}2^{-\ell|H|}2^{-(|H|-|U|)n_0} \\
&= \sum_{r=0}^{|H|} \binom{|H|}{r} S^r 2^{-\frac{h}{n^{k-2}}\gamma(k-1)}2^{-\ell|H|} \\
&= (2^{-\ell}(S+1))^{|H|}2^{-\frac{h}{n^{k-2}}\gamma(k-1)} \\
&\leq 2^{\frac{h}{2n^{k-2}}\gamma(k-1)+o\left(\frac{h}{n^{k-2}}\right)}2^{-\frac{h}{n^{k-2}}\gamma(k-1)} \\
&\leq 2^{-\frac{1}{2}\frac{h}{n^{k-2}}\gamma(k-1)+o\left(\frac{h}{n^{k-2}}\right)} \leq 2^{-\frac{h}{n^{k-1}}\gamma(k)}
\end{aligned}$$

for large enough n.

In the event $\left(\bigcup_\pi A_\pi\right)^c$, that is the event that for every π consistent with ρ, $\hbar((G|_\rho)|_\pi) \leq \sum_{i=1}^{k-1} h_i$, we can construct a decision tree for $G|_\rho$ of index-height at most $\sum_{i=1}^{k} h_i$ as follows: We first query all variables mentioning an index in H left unset by ρ. Then, at each branch π of the resulting tree which does not falsify any axiom of, we append a decision tree of minimum index-height representing $(G|_\rho)|_\pi$. $\square$

5.3 The lower bound for $\mathrm{RRef}_{d,N}$

Theorem 5.5. *For every integer $d > 0$, if F is an unsatisfiable CNF in n variables, N is an increasing function of n and s is a polynomial in n, every semantic depth d refutation of $\mathrm{RRef}_{d,N}(F, s)$ has size at least*

$$N^{\frac{1}{3}\left(\frac{\log s}{\log n}-2\right)^{\frac{1}{d-1}}}$$

for large enough n.

Proof. Let $h \coloneqq (1/3)(\log s/\log n - 2)^{1/(d-1)}$ and let $G_1, \ldots, G_t$ be a semantic depth d refutation of $\mathrm{RRef}_{d,N}(F, s)$ of size at most N^h. We assume that each G_i is either a literal or a disjunction of its immediate subformulas. Let A be a depth 1 subformula of some G_i. A is a 1-DNF or a 1-CNF formula, so applying Lemma 5.3 to it (or its negation respectively) with $k = 1$ and $p = N^{-1/2}$ and using as blocks $X_1, \ldots, X_r$ the variables in the depth 1 subformulas of $\mathrm{RRef}_{d,N}(F, s)$, we get, since $N^{-1/2} 3h \ln N = o\left(N^{-1/3}\right)$,

$$P[h(A|_\rho) > 3h] = o(N^{-h}).$$

Now, there are at most N^h depth 1 subformulas A in the refutation, hence, by Lemma 5.2 and the union bound, the probability that either there is a depth 1 subformula A with $h(A|_\rho) > 3h$ or $\mathrm{RRef}_{d,N}(F, s)|_\rho$ does not contain $\mathrm{RRef}_{d-1,N}(F, s)$ is $o(1)$. Therefore, for large n, there must be a restriction ρ_1' such that $\mathrm{RRef}_{d,N}(F, s)|_{\rho_1'}$ contains $\mathrm{RRef}_{d-1,N}(F, s)$ and all depth 1 subformulas of all $G_i|_{\rho_1'}$ are disjunctions or conjunctions of at most $3h$ literals. Let ρ_1 be a restriction extending ρ_1' such that $\mathrm{RRef}_{d,N}(F, s)|_{\rho_1}$ is exactly $\mathrm{RRef}_{d-1,N}(F, s)$. We continue by applying Lemma 5.3 with $k = 3h$ and $p = N^{-1/2}$ to a $3h$-CNF or $3h$-DNF depth 2 subformula B of $G_i|_{\rho_1}$ to get

$$P\left[h(B|_\rho) > (3h)^2\right] = o(N^{-h}).$$

Since $G_i|_{\rho_1}$ has at most N^h depth 2 subformulas, there is a restriction ρ_2 such that $\mathrm{RRef}_{d,N}(F, s)|_{\rho_1\rho_2}$ becomes $\mathrm{RRef}_{d-2,N}(F, s)$ and all depth 2 subformulas of all $G_i|_{\rho_1}$ can be represented by decision trees of height at most $(3h)^2$. A formula representable by a decision tree of height at most $(3h)^2$ can be written as both a $(3h)^2$-CNF and a $(3h)^2$-DNF, so for all $i \in [t]$, $G_i|_{\rho_1\rho_2} \in \Sigma_{d-1}^{N^h,(3h)^2}$.

Repeating the same argument $d-1$ times, we get restrictions $\rho_1, \ldots, \rho_{d-1}$ such that $\mathrm{RRef}_{d,N}(F, s)|_{\rho_1\cdots\rho_{d-1}}$ becomes $\mathrm{RRef}_{1,N}(F, s)$ and for all $i \in [t]$, $G_i|_{\rho_1\cdots\rho_{d-1}} \in \Sigma_2^{N^h,(3h)^{d-1}}$. We are now ready to apply Lemma 5.4. First notice that $\mathrm{RRef}_{1,N}(F, s)$ contains $\mathrm{R}^\ell\mathrm{Ref}(F, s)$ for large n, where $\ell \coloneqq (3h)^{d-1}$. For ρ selected randomly as specified in Lemma 5.4 for this ℓ, we get that the expected number of active indices is $s/2^\ell$, hence $\mathrm{RRef}_{1,N}(F, s)|_\rho$ contains $\mathrm{Ref}(F, s')$, where $s' \coloneqq s/2^{\ell+1}$, with high probability. Furthermore, Lemma 5.4 gives

$$P\left[\hbar(C|_\rho) > n^{(3h)^{d-1}}\right] \leq 2^{-\Omega(n)},$$

where C is a $(3h)^{d-1}$-DNF formula equivalent to some $G_i|_{\rho_1\cdots\rho_{d-1}}$. Therefore there must be a restriction ρ_d such that $\mathrm{RRef}_{d,N}|_{\rho_1\cdots\rho_d}$ becomes $\mathrm{Ref}(F, s')$ and for every

$i \in [t]$, $\hbar(G_i|_{\rho_1 \dots \rho_{d-1}}) \le n^{(3h)^{d-1}}$. Applying now the construction of Lemma 2.3[1] to $G_1|_{\rho_1 \dots \rho_{d-1}}, \dots, G_t|_{\rho_1 \dots \rho_{d-1}}$ gives a resolution refutation of $\mathrm{Ref}(F, s')$ of index-width at most $3n^{(3h)^{d-1}} = 3s/n^2$, contradicting Theorem 5.1 for large n. □

6 The main result

Theorem 6.1. *If* $\mathsf{P} \neq \mathsf{NP}$, *then depth* d *Frege systems are not automatable.*

Proof. Suppose there is an algorithm $\mathbf{A}$ which, given an unsatisfiable CNF formula G, returns a depth d refutation of G in time polynomial in $S(G)+S$, where $S(G)$ is the size of G and S the size of the smallest depth d refutation of G. Let $c, n_0 \ge 1$ be integers such that for every G with $|G| \ge n_0$, $\mathbf{A}$ runs in time at most $(S(G)+S)^c$. We will use $\mathbf{A}$ to decide in polynomial time whether 3-SAT is satisfiable. Given a 3-CNF formula F with n variables (and thus of size $O(n^3)$), we construct the formula $G := \mathrm{RRef}_{d,N}(F, s)$, where $s := n^{(3h)^{d-1}+2}$, $N := s$ and h is an integer such that

$$\left((3h)^{d-1}+2\right) h > c\left(\left((3h)^{d-1}+2\right)(2(d+3)) + 8\left((3h)^{d-1}+3\right)+1\right).$$

Notice that the left hand side of the above inequality is a polynomial of degree d in h and the right hand side a polynomial of degree $d-1$, hence such an h must exist. Since N and s are polynomials in n, the size of G is polynomial in n, hence its construction takes polynomial time. Let S be the size of the smallest depth d refutation of G and let $n_1 \ge n_0$ be an integer such that for all $n \ge n_1$,

$$\begin{aligned} F \text{ satisfiable} &\implies S + S(G) \le n^{\left((3h)^{d-1}+2\right)(2(d+3))+8\left((3h)^{d-1}+3\right)+1}; \\ F \text{ not satisfiable} &\implies S \ge n^{((3h)^{d-1}+2)h}. \end{aligned}$$

Here we use the bounds given by Proposition 4.1 and Theorem 5.5. To decide whether F is satisfiable, if $n < n_1$, then we check all possible assignments to its variables to see if there is a satisfying one. Otherwise, we run $\mathbf{A}$ on G for

$$n^{c\left(\left((3h)^{d-1}+2\right)(2(d+3))+8\left((3h)^{d-1}+3\right)+1\right)}$$

steps. If $\mathbf{A}$ stops, then we can assert that F is satisfiable; otherwise we can assert that F is unsatisfiable. □

7 Conclusion

We have shown the non-automatability result of bounded depth Frege system assuming $\mathsf{P} \neq \mathsf{NP}$. We do this, following [2], by constructing, given a CNF formula F, a formula $\mathrm{RRef}_{d,N}(F, s)$, and exhibiting a gap between the size of the shortest depth d Frege refutations of $\mathrm{RRef}_{d,N}(F, s)$ when F is satisfiable and the size of the shortest depth d Frege refutations of $\mathrm{RRef}_{d,N}(F, s)$ when F is not satisfiable.

[1] Lemma 2.3 is stated for height and width, but it is not hard to see that the same construction yields the lemma with index-height and index width instead of height and width respectively.

To show the lower bound for depth d Frege refutations of $\mathrm{RRef}_{d,N}(F,s)$ in the case F is not satisfiable, we employ the Furst-Saxe-Sipser switching lemma [8]. While sufficient for the purpose of showing non-automatability assuming $\mathsf{P} \neq \mathsf{NP}$, this can only give lower bounds of the form n^h, where h is a barely superconstant function of n. It would be nice to have an exponential lower bound. In particular, as in [2], an exponential lower bound would rule out the automatability of bounded depth Frege systems in quasipolynomial time unless NP problems can be solved in quasipolynomial time, and their automatability in subexponential time unless NP problems can be solved in subexponential time.

$\mathrm{RRef}_{d,N}(F,s)$ consists of formulas of depth d. In particular, this does not preclude the possibility of bounded depth Frege systems being automatable on refuting, say CNF formulas. A natural question is whether we could use CNFs, or at least formulas of constant depth, not depending on d, instead. Let us mention here that whether there is a constant depth formula exponentially separating depth d from $d+1$ Frege is open as well; currently, only a super-polynomial separation is known [13] (see also [15, Section 14.5]). Moreover, the formulas $\mathrm{RRef}_{d,N}(F,s)$ are ad hoc and rather artificial. It would be nice if one could establish a lower bound for formulas $\mathrm{Ref}_d(F,s)$ for an unsatisfiable formula F, encoding the fact that there are depth d refutations of F of size s (see Problem 2 in [16]), showing that proving lower bounds for a depth d Frege system is hard within the system. The latter problem for a proof system is considered by Pudlák [16] to be a more important question than the question of whether the system is automatable. Note that a CNF encoding of $\mathrm{Ref}_d(F,s)$ is a candidate formula for the question of whether bounded depth Frege systems for refuting CNFs are automatable, and a CNF encoding of the reflection principle $\mathrm{Sat}(F,v) \wedge \mathrm{Ref}_d(F,s)$, where $\mathrm{Sat}(F,v)$ encodes that v is an assignment satisfying F, is a candidate formula for the depth d vs depth $d+1$ Frege problem (see [16]).

Finally, the non-automatability result of [2] has been shown for cutting planes [11], $\mathrm{Res}(k)$ [10], and various algebraic proof systems [7]. As far as we know, a remaining open case is sum of squares.

Acknowledgements

I would like to thank Alexander Razborov for numerous remarks and suggestions that greatly improved the presentation of the paper.

References

[1] Michael Alekhnovich and Alexander A. Razborov. Resolution is not automatizable unless W[P] is tractable. *SIAM Journal of Computing*, 38:1347–1363, 2008.

[2] Albert Atserias and Moritz Müller. Automating resolution is NP-hard. *Journal of the ACM*, 67:31:1–31:17, 2020.

[3] Arnold Beckmann and Samuel Buss. Separation results for the size of constant-depth propositional proofs. *Annals of Pure and Applied Logic*, 136:30–55, 2005.

[4] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth frege proofs. *Computational Complexity*, 13:47–68, 2004.

[5] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On interpolation and automatization for frege systems. *SIAM Journal of Computing*, 29:1939–1967, 2000.

[6] Stefan Dantchev and Søren Riis. On relativisation and complexity gap. In *Proceedings of the 12th Annual Conference of the EACSL*, pages 142–154, 2003.

[7] Susanna de Rezende, Mika Göös, Jakob Nordström, Toniann Pitassi, Robert Robere, and Dmitry Sokolov. Automating algebraic proof systems is NP-hard. In *Proccedings of the 53rd Annual ACM Symposium on Theory of Computing*, pages 209–222, 2021.

[8] Merrick Furst, James Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[9] Michal Garlík. Resolution lower bounds for refutation statements. In *Proccedings of the 44th International Symposium on Mathematical Foundations of Computer Science*, volume 138, pages 37:1–37:13, 2019.

[10] Michal Garlík. Failure of feasible disjunction property for k-DNF resolution and NP-hardness of automating it. *Electronic Colloqium on Computational Complexity*, 2020.

[11] Mika Göös, Sajin Koroth, Ian Mertz, and Toniann Pitassi. Automating cutting planes is np-hard. In *Proccedings of the 52nd Annual ACM Symposium on Theory of Computing*, pages 68–77, 2020.

[12] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 6–20, 1986.

[13] Russell Impagliazzo and Jan Krajícek. A note on conservativity relations among bounded arithmetic theories. *Mathematical Logic Quarterly*, 48:375–377, 2002.

[14] Jan Krajíček. Lower bounds to the size of constant-depth propositional proofs. *Journal of Symbolic Logic*, 59:73–86, 1994.

[15] Jan Krajíček. *Proof Complexity.* Cambridge University Press, 2019.

[16] Pavel Pudlák. Reflection principles, propositional proof systems, and theories, 2020. arXiv:2007.14835.

[17] Nathan Segerlind, Samuel Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM Journal of Computing*, 33:1171–1200, 2004.

[18] Michael Sipser. Borel sets and circuit complexity. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing*, pages 61–69, 1983.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il