Exponential Lower Bounds for AC^0-Frege Imply Superpolynomial Frege Lower Bounds

YUVAL FILMUS and TONIANN PITASSI, University of Toronto
RAHUL SANTHANAM, University of Edinburgh

We give a general transformation which turns polynomial-size Frege proofs to subexponential-size AC^0-Frege proofs. This indicates that proving truly exponential lower bounds for AC^0-Frege is hard, since it is a longstanding open problem to prove super-polynomial lower bounds for Frege. Our construction is optimal for proofs of formulas of unbounded depth.

As a consequence of our main result, we are able to shed some light on the question of automatizability for bounded-depth Frege systems. First, we present a simpler proof of the results of Bonet et al. showing that under cryptographic assumptions, bounded-depth Frege proofs are not automatizable. Second, we show that because our proof is more general, under the right cryptographic assumptions, it could resolve the automatizability question for lower depth Frege systems.

Categories and Subject Descriptors: F.1.3 [**Computation by abstract devices**]: Complexity measures and classes—*Relations among complexity classes*

General Terms: Theory

Additional Key Words and Phrases: Proof complexity

ACM Reference Format:
Yuval Filmus, Toniann Pitassi and Rahul Santhanam, 2013. Exponential lower bounds for AC^0-Frege imply superpolynomial Frege lower bounds. *ACM Trans. Comput. Theory* V, N, Article A (January YYYY), 17 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Proof complexity was introduced by [Cook and Reckhow 1979] as a framework within which to study the NP vs. coNP problem. Cook and Reckhow defined *propositional proof systems* in a very general way by insisting only that proofs be verifiable in polynomial time, and showed that the existence of a propositional proof system in which all tautologies have polynomial-size proofs is equivalent to NP = coNP. They suggested a program to separate NP and coNP (and thereby P and NP) by showing superpolynomial proof size lower bounds for explicit tautologies in progressively stronger proof systems. The hope was that techniques from logic and proof theory could be effective where techniques inspired by recursion theory or combinatorics are not. The fact that the very definition of the P vs. NP question involves the notion of "proof" in a fundamental way makes this hope somewhat plausible. Indeed, over the past couple

Supported by NSERC. This material is based upon work supported by the National Science Foundation under agreement No. DMS-1128155. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Authors' addresses: Yuval Filmus, (Current address) Institute for Advanced Study, Princeton; Toniann Pitassi, University of Toronto; Rahul Santhanam, University of Edinburgh.

DOI: http://dx.doi.org/10.1145/0000000.0000000

of decades, lower bounds have been shown for various natural proof systems [Haken 1985; Beame et al. 1992]. However, lower bounds for natural systems such as Frege and Extended Frege still seem out of reach.

In this paper, we draw a connection between two fundamental questions in proof complexity. The first question is to prove strong lower bounds for bounded-depth Frege. Superpolynomial lower bounds are known for this proof system, but there aren't any lower bounds known that are purely exponential, i.e., $2^{\Omega(n^c)}$ where the constant c doesn't depend on the depth of lines in the proof (the best known lower bound is $\Omega(2^{n^{5^{-d}}})$ for depth d Frege [Beame et al. 1992; Pitassi et al. 1993; Krajíček et al. 1995; Fu and Urquhart 1996]). The second question, which is perhaps the major open question in proof complexity, is to obtain superpolynomial lower bounds for Frege. This question is believed to be very hard — it is non-trivial even to think of plausible candidate tautologies for which superpolynomial lower bounds are believed to hold [Bonet et al. 1995; Krajíček 2011]. We show that progress on the first question would lead to progress on the second, by giving a general simulation of polynomial-size Frege proofs by subexponential-size bounded-depth Frege proofs. More precisely, we show that even a $2^{n^{\omega(1/d)}}$ proof size lower bound for proving CNF tautologies in depth d Frege would translate to a superpolynomial proof size lower bound for Frege.

The proof of this connection is inspired by a result in circuit complexity, further strengthening the "mapping" between proof complexity and circuit complexity. The circuit complexity result we draw inspiration from is that NC^1 can be simulated by bounded-depth circuits with subexponential size [Allender et al. 2008]. The standard proof of this goes via a divide-and-conquer technique. We use a similar technique in our context, however our task is made harder in a sense by the fact that we need to reason within bounded-depth Frege about equivalence of various alternative representations of a function. The technical heart of our proof involves such reasoning.

Our result is also relevant to algorithmic analysis, which is another major motivation for studying proof complexity. A propositional proof system can be thought of as a non-deterministic algorithm for deciding if a formula is a tautology or not. Proof systems such as bounded-depth Frege and Frege provide particularly simple and natural examples of such algorithms. Indeed, many of the algorithms and heuristics used in practice for solving SAT, such as DPLL and Clause Learning, arise from *determinizing* the non-deterministic algorithm corresponding to some natural proof system [Pipatsrisawat and Darwiche 2011]. Thus lower bounds for proof systems give us information on the performance of algorithms used in practice.

Algorithmic analysis would appear to be a simpler question than proving complexity lower bounds, since a complexity lower bound is a statement about *any* possible algorithm for a problem, while algorithmic analysis deals with specific algorithms. There are somewhat artificial algorithms such as Levin's optimal algorithm for SAT [Levin 1973] whose analysis is just as difficult as proving complexity lower bounds. However, one might hope that for more natural algorithms, such as those corresponding to natural propositional proof systems, this is not the case. Our current lack of progress in proving proof complexity lower bounds indicates that there might be barriers even in algorithmic analysis of natural algorithms. Our main result here can be interpreted as saying that the algorithmic analysis question for the algorithm corresponding to bounded-depth Frege is as hard as the question for the algorithm corresponding to Frege (which in some sense is a more sophisticated algorithm). In general, it would be useful to have a theory of algorithmic analysis which gives us information about the relative difficulty of analyzing various natural algorithms. We make a small step in this direction in the setting of non-deterministic algorithms recognizing TAUT, the set of all tautologies.

There are a couple of interesting byproducts of our main result. First, we are able to prove *tight* bounds for proving certain explicit tautologies in bounded-depth Frege. Lower bounds for the tautologies we consider were already shown in [Krajíček 1994]. We give corresponding upper bounds as a corollary of our simulation of Frege by bounded-depth Frege.

Second, we address the question of *automatizability* for bounded-depth Frege systems. A proof system $\mathcal{P}$ is automatizable if there is an algorithm that given a tautology f outputs a $\mathcal{P}$-proof of f in time polynomial of the smallest $\mathcal{P}$-proof of f. Despite considerable effort, the question of whether low-depth proof systems are automatizable is unresolved. Bonet, Domingo, Gavaldà, Maciel and Pitassi [2004] show that depth k Frege systems are not automatizable under a cryptographic assumption, but their result breaks down for small k (less than 6). We use our main result to re-derive their main theorem. Our proof is cleaner and simpler than theirs, and we show that it could potentially resolve the automatizability question for lower depth Frege systems than what is currently known.

Subsequent to our work, two alternative proofs of our main theorem have appeared. Müller's proof [Müller 2013] uses model-theoretic methods, and Cook and Ghasemloo's proof [Ghasemloo and Cook 2013] uses bounded arithmetic. In both proofs, the role of the circuit complexity result is played by Nepomnjaščij's theorem [Nepomnjaščij 1970], which states that $\mathrm{NTimeSpace}(n^{O(1)}, n^{o(1)}) \subseteq \mathrm{AltTime}(O(1), O(n^{\epsilon}))$ for every $\epsilon > 0$. Cook and Ghasemloo's proof also applies for the uniform case: it shows that uniform polynomial-size Frege proofs translate to uniform subexponential-size bounded-depth Frege proofs.

Paper organization. After describing some necessary background in Section 2, we formally state our main theorem in Section 3. Section 3.1 shows that our simulation is tight, and in Section 3.2 we prove that Frege systems do not have feasible interpolation and are not automatizable unless the Diffie–Hellman problem is computable by polynomial-size circuits, thus reproving the main result of [Bonet et al. 2000]. The proof of our main theorem constitutes Section 4.

2. PROOF SYSTEMS

We will work with the propositional sequent calculus, PK. In the fundamental work of Cook and Reckhow [1979], many reasonable formulations of Frege systems (including *all* PK-like systems) were studied and shown to be polynomially equivalent; we work with PK for convenience, but any other Frege system would do.

Each line in a PK proof is a *sequent* of the form $A_1, \ldots, A_k \longrightarrow B_1, \ldots, B_m$, where $\longrightarrow$ is a new symbol and A_i, B_j are formulas. The intended meaning is that the conjunction of the A_i's implies the disjunction of the B_j's.

A PK *proof* of $\longrightarrow f$ is a sequence of sequents, such that each sequent is either an instance of the axiom $A \longrightarrow A$, or follows from previous sequents from one of the inference rules, and such that the final sequent is $\longrightarrow f$.

The rules of PK are of three types: (i) the structural rules, (ii) the logical rules, and (iii) the cut rule.

The structural rules are weakening (formulas can always be added to the left or to the right), contraction (two copies of the same formula can be replaced by one), and permutation (formulas in a sequent can be reordered).

The cut rule allows deriving $\Gamma \longrightarrow \Delta$ from $A, \Gamma \longrightarrow \Delta$ and $\Gamma \longrightarrow A, \Delta$. The formula A is called the *cut formula*.

The logical rules, shown below, allow us to introduce each connective on both the left side and the right side.

1. (Negation Left, $\neg$**L**) From $\Gamma \longrightarrow A, \Delta$, derive $\neg A, \Gamma \longrightarrow \Delta$.
2. (Negation Right, $\neg$**R**) From $A, \Gamma \longrightarrow \Delta$, derive $\Gamma \longrightarrow \neg A, \Delta$.
3. (And Left, $\wedge$**L**) From $A, B, \Gamma \longrightarrow \Delta$, derive $A \wedge B, \Gamma \longrightarrow \Delta$.
4. (And Right, $\wedge$**R**) From $\Gamma \longrightarrow A, \Delta$ and $\Gamma \longrightarrow B, \Delta$, derive $\Gamma \longrightarrow A \wedge B, \Delta$.
5. (Or Left, $\vee$**L**) From $A, \Gamma \longrightarrow \Delta$ and $B, \Gamma \longrightarrow \Delta$, derive $A \vee B, \Gamma \longrightarrow \Delta$.
6. (Or Right, $\vee$**R**) From $\Gamma \longrightarrow A, B, \Delta$ derive $\Gamma \longrightarrow A \vee B, \Delta$.

When presenting proofs in PK, we will only mention the logical rules and the cut rule, but not the structural rules.

The *size of a formula* is the total number of symbols occuring in it, and the *size of a PK proof* is the sum of the sizes of all formulas occurring in the proof.

The following two definitions are taken from [Krajíček 1995, Definition 4.3.1].

We can view every formula φ over the de Morgan basis $\{\wedge, \vee, \neg\}$ as an improper binary tree, in which nodes corresponding to subformulas of the form $\neg\psi$ have only one child ψ. The *logical depth* of a formula φ, denoted by $\mathrm{ldp}(\varphi)$, is the depth of the formula when considered as an improper binary tree. For example, $(A \wedge B) \wedge C$ has logical depth 2, and $\neg A$ has logical depth 1. A formula whose logical depth is D has size at most $2^{D+1} - 1$, and can depend on at most 2^D variables.

Given a formula φ and a leaf f in the improper binary tree corresponding to φ, consider the root-to-leaf path leading to f, and write out all the connectives showing up on the path in order, making up the *connective sequence* of f. We can write this sequence as a sequence of "runs" of identical connectives. The *connective depth* of f is the number of runs in its connective sequence. The *depth* of a formula is the maximal connective depth of a leaf. For example, consider the formula $(A \wedge (B \vee (C \vee (\neg\neg D))) \wedge E$ and the leaf D. The connective sequence of D is $\wedge \wedge \vee \vee \neg\neg$, which consists of three runs, and so the connective depth of D (and the depth of the entire formula) is 3.

We have given definitions of two different notions of depth. We will use logical depth to reason about formulas in Frege proofs, and depth to reason about formulas in bounded depth proofs.

A *cut-depth* k proof, also called an AC^0_k*-Frege* proof, is a PK proof in which every *cut formula* in the proof has depth at most k (other formulas are allowed to have arbitrary depth). Note that in the literature, an AC^0_k-Frege proof is often defined to be a PK proof where *all* formulas have depth at most k. This definition is equivalent to ours if the proven formula has depth at most k.

For technical reasons, we will need all the formulas in our proofs to be balanced (have depth logarithmic in their size). By the following result of Reckhow, this can be assumed without loss of generality for polynomial-size proofs.

THEOREM 2.1 ([RECKHOW 1976], [KRAJÍČEK 1995, LEMMA 4.4.14]). *If a formula of logical depth D has a PK proof of size s, then it has a PK proof of size $s^{O(1)}$ in which all formulas have logical depth $D + O(\log s)$.*

The proof of Reckhow's theorem is based on Spira–Brent-style balancing [Spira 1971; Brent 1974].

We briefly define feasible interpolation and automatizability for proof systems. We comment that while the definition of feasible interpolation first appears explicitly in [Bonet et al. 2000], the concept had been introduced to proof complexity by Jan Krajíček [1997].

Definition 2.2 (*Bonet, Pitassi and Raz [2000]*). A proof system S has feasible interpolation if for every sequence of tautologies of the form $F_n = A_n(\vec{x}, \vec{y}) \vee B_n(\vec{x}, \vec{z})$ which have proofs in S of size $poly(n)$, where $\vec{y}$ and $\vec{z}$ are disjoint sets of variables, there is a sequence of polynomial-size circuits C_n such that for any truth assignment α to $\vec{x}$, $C_n(\alpha) = 0$ implies that $B_n(\alpha, \vec{z})$ is a tautology, and $C_n(\alpha) = 1$ implies that $A_n(\alpha, \vec{y})$ is a tautology.

Definition 2.3 (*Bonet, Pitassi and Raz [2000]*). A proof system S is *automatizable* if there exists an algorithm A such that for all tautologies f, $A(f)$ returns an S-proof of f, and the runtime of A on f is polynomial in the size of the smallest S-proof of f.

Alekhnovich and Razborov [2001] showed that Resolution is not automatizable under a parameterized complexity assumption. Bonet, Pitassi and Raz [2000] showed that automatizability implies feasible interpolation, and also proved that TC^0-Frege (and hence Frege) does not have feasible interpolation under the assumption that the Diffie–Hellman function does not have polynomial-size circuits. Bonet et al. [2004] built on this result to show that bounded-depth Frege does not have feasible interpolation if the Diffie–Hellman function does not have subexponential-size circuits. We reprove this result in Section 3.2.

3. MAIN THEOREM AND APPLICATIONS

Our main result is the following theorem.

THEOREM 3.1. *Let φ be a formula provable in Frege in size s, satisfying* $\mathrm{ldp}(\varphi) \leq C \log s$*. For every $k \geq 1$ there is an AC^0_{k+4}-Frege proof of φ of size $2^{O(ks^{O(C/k)})}$.*

COROLLARY 3.2. *Let φ be a formula of size s and logical depth at most $C \log s$, and let c be an integer. If φ has a Frege proof of size $O(s^\ell)$ then for every $k \geq 1$ there is an AC^0_{k+4}-Frege proof of φ of size $2^{O(ks^{O_\ell(C/k)})}$.*

The proof of the theorem occupies Section 4. We comment that sometimes depth is defined without counting negations, and in that case the proofs constructed in Theorem 3.1 have depth $k+3$ rather than $k+4$. It might be possible that 4 can be replaced by a slightly smaller integer.

3.1. Tightness of our simulation

The result analogous to Corollary 3.2 for circuit complexity shows that any function computable by a polynomial-size formula can be computed by depth d circuits of size $\exp(n^{O(1/d)})$. This result is tight, since Håstad's theorem [Håstad 1987] proves that the parity function on n Boolean variables requires AC^0_d circuits of size $\exp(n^{1/d})$.

Similarly we can show that our result is also tight. The following theorem states that there are formulas that have polynomial-size Frege proofs, but that require AC^0_d proofs of size exponential in $n^{1/d}$.

THEOREM 3.3. *For every d there is a sequence of balanced formulas φ_n of depth $d+3$ provable in Frege by a proof of size s_n such that every AC^0_{d-1} proof of φ_n requires size $2^{s_n^{\Omega(1/d)}}$.*

PROOF. The formula φ_n is PHP_n, the pigeonhole principle with $n+1$ pigeons and n holes, with each variable replaced by a Sipser function [Sipser 1983] of depth d. Buss [1987] showed how to prove PHP_n using a Frege proof of size $n^{O(1)}$. Substituting the Sipser functions, we obtain a Frege proof of size $n^{d+O(1)}$.

Conversely, Krajíček [1994, §4] gives a lower bound of $\exp(n^{1/5})$ for proving φ_n in tree-like AC^0_d (a tree-like proof is one in which each sequent is used at most once). Since an arbitrary AC^0_{d-1} proof can be simulated in tree-like AC^0_d with at most a quadratic blow-up [Krajíček 1994, Proposition 1.1], this gives a lower bound of $\exp(n^{1/5})$ for proving φ_n in AC^0_{d-1}. □

Since the formulas φ_n are balanced, Theorem 3.1 applies, and with $k = d - 5$, gives proofs essentially matching the lower bound.

The above result proves tightness for formulas of high depth. We conjecture that our simulation is also tight with respect to CNF formulas. The obvious formula for witnessing the lower bound is the pigeonhole principle itself. However, as an artifact of the switching lemma technique used to obtain depth d Frege lower bounds for the pigeonhole principle, the current best lower bound is exponential in n^{ϵ^d} for suitable $\epsilon > 0$ [Beame et al. 1992; Fu and Urquhart 1996]. It is a well-known open problem to improve the lower bound to $\exp(n^{1/d})$ for the pigeonhole principle, or for any other CNF formula. Such a result would show that our simulation is tight even for CNF formulas.

3.2. Automatizability

Using our theorem, we are able to show that bounded-depth Frege is not automatizable, under an assumption about the hardness of factoring. While this result has already been known [Bonet et al. 2004], we show how to prove it as a simple corollary of our main theorem.

The starting point is a similar result for Frege.

THEOREM 3.4 ([BONET ET AL. 2000]). *Frege systems do not have feasible interpolation and are not automatizable unless the Diffie–Hellman problem is computable by polynomial-size circuits.*

The Diffie–Hellman problem is based on a prime number p, $|p| = n$. The input to the problem is a number g less than p, and numbers $g^a \pmod{p}$, $g^b \pmod{p}$, for some numbers $a, b \leq p$. The output should be $g^{ab} \pmod{p}$. The main lemma from [Bonet et al. 2000] shows that a particular tautology, DH_p, stating that the Diffie–Hellman function is well-defined, has Frege proofs of size $O(|p|^c)$, where $c \leq 4$. Our normal form theorem shows that DH_p has AC^0_k-Frege proofs of size $2^{O(kn^{O(1/k)})}$. Hence it follows that if AC^0_k-Frege is automatizable (or has feasible interpolation) then the Diffie–Hellman problem can be solved in time $2^{O(kn^{O(1/k)})}$ for all k. This implies the following theorem.

THEOREM 3.5 ([BONET ET AL. 2004]). *Bounded-depth Frege systems do not have feasible interpolation and are not automatizable unless for all δ, the Diffie–Hellman problem is computable by circuits of size 2^{n^δ}.*

Unfortunately, the quality of this negative result degrades for small k. Indeed, despite considerable effort, it is unknown whether or not very low depth Frege systems (when k is less than 5) are automatizable (the recent paper [Atserias and Maneva 2011] reveals a connection between automatizability of AC^0_2-Frege with bottom fan-in 2 and feasibility of mean-payoff games). The main reason for this is that the Diffie–Hellman function is not hard enough! Algorithms exist for computing discrete log over all finite fields, and hence for Diffie–Hellman, that run in time $\exp(\sqrt{n})$. Moreover, the number field sieve is conjectured to solve discrete log (and thus Diffie–Hellman) in time $\exp(\sqrt[3]{n})$. Other algorithms for discrete log over small characteristic are conjectured to run even faster: Joux's algorithm [Joux 2013] in time $\exp(\sqrt[4]{n})$, and the BGJT algorithm [Barbulescu et al. 2013] in quasipolynomial time.

On the other hand, it seems entirely possible to come up with a different interpolant statement for another function that is much harder – truly exponential in n, and that still has efficient Frege proofs. Using our main theorem (which scales down *any* Frege proof), this would imply new negative results for automatizability and feasible interpolation for lower depth Frege systems than what is currently known.

4. PROOF OF MAIN THEOREM

4.1. Proof overview

Suppose that P is a Frege proof of some formula f. We want to simulate P by a subexponential-size depth d Frege proof of f. The high-level idea behind the simulation is to replace every formula in the proof by its equivalent depth d (subexponential-size) *flattened* formula, and then to show that if C was derived by a rule from A and B, then the flattened version of C can be efficiently derived from the flattened versions of A and B.

We can assume without loss of generality that all formulas f in the proof are balanced (Reckhow's theorem). We first review the translation of a balanced formula f to its flattened form. Suppose that we want to replace f, of size n and logical depth $\log n$, by a depth 4 formula. The idea is to view f as consisting of two layers: the top layer is a formula, f_1, of height $(\log n)/2$, and the bottom layer consists of $2^{(\log n)/2} = \sqrt{n}$ subformulas, $g_1, \ldots, g_{\sqrt{n}}$, each of height $(\log n)/2$. Since f_1 has height $(\log n)/2$, it has at most $\sqrt{n}$ inputs, and thus can be written as either a CNF or a DNF formula (of its inputs) of size $\sqrt{n}2^{\sqrt{n}}$. Similarly, each formula in the bottom layer can be written as either a CNF or a DNF formula of size $\sqrt{n}2^{\sqrt{n}}$. Writing f_1 as a CNF formula, and writing all formulas g_j in the bottom layer as DNF formulas, we obtain a new formula for f of depth 4 and total size $O(n2^{2\sqrt{n}})$. (The depth is 4 because we can merge the middle two AND layers to obtain the following layer structure: OR, AND, OR, NOT.) In a similar manner, we can replace any formula f, of size n and logical depth $\log n$, by a depth $d+2$ formula: Now we break f up into d equally-spaced layers, each of size $(\log n)/d$. Again, we write the formula at the top layer as a CNF formula, the formulas at the next layer as DNF formulas, and so on. This gives a formula of depth $2d+1$ and total size $O(n2^{dn^{1/d}})$, but since we alternated CNF/DNFs, we can collapse every other layer to obtain a new flattened formula of depth $2d-(d-1)+1 = d+2$.

Now that we have flattened translations of each formula in P, it remains to fill in the proof, to show that the flattened versions can be derived from one another. In order to carry this out, we define a more general procedure for flattening a formula as follows. Let $\vec{d}$ be any *depth vector* – i.e., a sequence of increasing numbers, where each number in the sequence is between 1 and $\log n$. Then from a balanced formula f of size n and logical depth $\log n$, $\vec{d}$ defines a new flattened formula of depth $|\vec{d}|+3$: we break f up into $|\vec{d}|+1$ many layers, where now instead of the layers being equally spaced, the breakpoints are specified by $\vec{d}$. For example, if $\vec{d} = (4,12)$ and f has depth 20, then the $\vec{d}$-flattened version of f will have 3 layers, the top layer containing levels 1 through 3, the second level 4 through 11, and the third level 12 through 20. Our main lemma shows that for any balanced formula f and any two depth vectors $\vec{d_1}$, $\vec{d_2}$, there are efficient low-depth Frege proofs showing that the $\vec{d_1}$-flattened version of f is equivalent to the $\vec{d_2}$-flattened version of f. This main lemma will then allow us to prove that for any rule of our proof system, the flattened versions of the antecedent formulas derive the flattened version of the consequent formula.

4.2. Reducing formula depth

As described in the overview, we reduce the depth of a formula using a divide-and-conquer technique. The idea is to decompose the formula into relatively small subtrees, and replace each sub-tree by a CNF or DNF which is equivalent to the formula computed by the sub-tree.

Definition 4.1. Let φ be an arbitrary formula depending on n variables. Denote by $\mathrm{CNF}(\varphi)$ ($\mathrm{DNF}(\varphi)$) some canonically chosen CNF (DNF) representing φ of size $O(n2^n)$. We require that $\mathrm{CNF}(p \land q) = \mathrm{DNF}(p \land q) = p \land q$, and similarly for $p \lor q$ and $\neg p$, when p and q are variables.

We think of formulas as trees in which internal nodes are either binary (if the corresponding connective is $\land$ or $\lor$) or unary (when the connective is $\neg$), and leaves are labelled by variables. Each formula has an equivalent formula of the same size where negations only appear immediately above leaves, just by applying de Morgan's laws repeatedly to "move" negations down. We will say that such formulas are in *negation normal form*, and will work with such formulas throughout our simulation.

Definition 4.2. A formula is in *negation normal form* if negations only appear next to variables, and there are no double negations. Let φ be a formula in negation normal form. Its *dual* form $\mathrm{M}(\varphi)$ is obtained from φ by switching $\land$ and $\lor$ and negating all literals, that is for each variable x switching x and $\neg x$.

Note that $M(\varphi)$ is logically equivalent to $\neg\varphi$ by de Morgan's laws.

We define two *canonical flattened forms* in parallel. We stress that these forms apply to *arbitrary* formulas, which need not be in negation normal form.

Definition 4.3. Let $\vec{d} = d_1, \ldots, d_k$ be a vector of increasing positive integers. The *conjunctive flattened form* $\mathrm{C}(\varphi; \vec{d})$ and *disjunctive flattened form* $\mathrm{D}(\varphi; \vec{d})$ of a formula φ are defined recursively as follows. If $k = 0$ (i.e., $\vec{d}$ is the empty vector) or $d_1 \geq \mathrm{ldp}(\varphi)$ then $\mathrm{C}(\varphi; \vec{d}) = \mathrm{CNF}(\varphi)$ and $\mathrm{D}(\varphi; \vec{d}) = \mathrm{DNF}(\varphi)$. Otherwise, let ψ be the formula obtained from φ by trimming the tree at logical depth d_1. The formula ψ depends on the variables of φ as well as on variables corresponding to subformulas of φ at logical depth d_1; we call these *true variables* and *subformula variables*, respectively. Let v_χ denote the subformula variable corresponding to the subformula χ.

We explain how to calculate the conjunctive flattened form; the disjunctive flattened form is analogous. Start with $\mathrm{CNF}(\psi)$. Let $\vec{e} = d_2 - d_1, \ldots, d_k - d_1$. Replace each positive occurrence of a subformula variable v_χ in $\mathrm{CNF}(\psi)$ with $\mathrm{D}(\chi; \vec{e})$, and each negative occurrence with $\mathrm{M}(\mathrm{C}(\chi; \vec{e}))$. The result is $\mathrm{C}(\varphi)$.

The flattened forms are both shallow and not too large.

Definition 4.4. Let φ be a formula and $\vec{d} = d_1, \ldots, d_k$ be a vector of increasing positive integers, such that $d_1 \leq \mathrm{ldp}(\varphi)$. Let $d_0 = 0$ and $d_{k+1} = \mathrm{ldp}(\varphi)$. The *extent* of φ with respect to $\vec{d}$ is

$$\mathrm{ex}(\varphi; \vec{d}) = \max\{d_{i+1} - d_i : 0 \leq i \leq k\}.$$

LEMMA 4.5. *Let φ be a formula and $\vec{d}$ a vector of length k and extent $x = \mathrm{ex}(\varphi; \vec{d})$. Then $\mathrm{C}(\varphi; \vec{d})$ and $\mathrm{D}(\varphi; \vec{d})$ are formulas of depth at most $k + 3$ and size $2^{O(k2^x)}$ equivalent to φ.*

PROOF. It is easy to see, using de Morgan's laws, that the flattened forms are equivalent to the original formula. The recursive definition of the flattened forms ensures that all negations are pushed to the leaves, and that CNFs and DNFs alternate. Therefore their depth is $k + 3$ (the depth of a CNF/DNF is 3).

In order to estimate the size, denote by $M(k, x)$ the maximum size of a flattened form of a formula with respect to a vector of length k and extent x. By Definition 4.1,

$M(0,x) = O(2^x 2^{2^x}) = 2^{O(2^x)}$, since a formula of logical depth x depends on at most 2^x variables. Since $M(0,x)$ also bounds the number of literals in a CNF/DNF, $M(k{+}1,x) \leq M(0,x) + M(0,x)M(k,x)$. Therefore $M(k,x) \leq \sum_{l=0}^{k} M(0,x)^{l+1} = 2^{O(k2^x)}$. □

4.3. Brute force proof techniques

We state some simple lemmas which will enable us to reason about flattened forms.

We start with a general proof technique for arbitrary sequents.

Definition 4.6. A *truth assignment* for variables $x_1,\dots,x_n$ is a function $f\colon \{x_1,\dots,x_m\} \longrightarrow \{\bot,\top\}$ assigning to each variable a truth value ($\bot$ is False, $\top$ is True).

LEMMA 4.7. *Let φ be a formula of size m depending upon the set of variables X of size n, and consider a truth assignment f for X. If φ is satisfied by f then the sequent*

$$\{x \in X : f(x) = \top\} \longrightarrow \{x \in X : f(x) = \bot\}, \varphi$$

has a cut-free proof of size $O(nm(n+m))$. If φ is falsified by f, then the same is true for the sequent

$$\{x \in X : f(x) = \top\}, \varphi \longrightarrow \{x \in X : f(x) = \bot\}.$$

Furthermore, if φ is in negation normal form then negation rules are only applied to axioms $x \longrightarrow x$.

PROOF. The proof is by structural induction. Denote by $S(\varphi)$ the sequent alluded to in the statement of the lemma. We first describe the proof, and then analyze its size.

If $\varphi = x$ is a variable then $S(\varphi)$ follows from the axiom $x \longrightarrow x$ using weakening. If $\varphi = \neg\psi$ then $S(\varphi)$ follows from $S(\psi)$ by using the appropriate $\neg$ introduction rule. When φ is in negation normal form, this case can happen only when ψ is a variable x, and so we can obtain $S(\varphi)$ from $x \longrightarrow x$ by using the appropriate $\neg$ introduction rule followed by weakening.

If $\varphi = \psi \wedge \chi$ and φ is satisfied by f, then $S(\varphi)$ follows from the sequents $S(\psi)$ and $S(\chi)$ by using the right $\wedge$ introduction rule. If it is falsified by f, then either ψ is falsified or χ is falsified. Suppose, without loss of generality, that ψ is falsified. Then $S(\varphi)$ follows from $S(\psi)$ by using the left $\wedge$ introduction rule. The proofs are similar if the main connective is $\vee$ instead of $\wedge$.

In total, we have eliminated each connective by using one logical rule, and each variable using n weakening rules. The total number of sequents needed is therefore $O(nm)$, each of size at most $n+m$. □

LEMMA 4.8. *Let $\Gamma \longrightarrow \Delta$ be a sequent of size m depending on the set of variables X of size n. Suppose that the sequent is valid under some truth assignment f. Then the sequent*

$$\Gamma, \{x \in X : f(x) = \bot\} \longrightarrow \Delta, \{x \in X : f(x) = \top\}$$

has a cut-free proof of size $O(nm(n+m))$.

Furthermore, if all formulas in Γ, Δ are in negation normal form then negation rules are only applied to axioms $x \longrightarrow x$.

PROOF. Since the sequent is valid under f, either one of the formulas in Γ is false, or one of the formulas in Δ is true. Use Lemma 4.7 to prove the corresponding sequent, and conclude the sequent in the statement by using at most m weakening rules, for an extra size of $O(m(n+m))$. □

LEMMA 4.9. *Let $\Gamma \longrightarrow \Delta$ be a valid sequent of size m, in which n variables appear. The sequent is provable using a proof of size $O(m^2 n 2^n)$ which cuts only on variables.*

Furthermore, if all formulas in Γ, Δ are in negation normal form then negation rules are only applied to axioms $x \longrightarrow x$.

PROOF. Let X be the set of variables appearing in $\Gamma \longrightarrow \Delta$; note that $n = |X| \leq m$. Apply Lemma 4.8 for each of the 2^n truth assignments. Divide all truth assignments into pairs where only the value of the leftmost variable in $x \in X$ differs. Apply the cut rule to all pairs, eliminating the variable x. Continue this way, eliminating all variables in order, to obtain a proof of $\Gamma \longrightarrow \Delta$. In total, the proof uses $2^n - 1$ cuts. □

Our next lemma states that we can substitute formulas for variables to get a valid proof.

LEMMA 4.10. *Let π be a proof of $\Gamma \longrightarrow \Delta$ of size s, let $x_1, \ldots, x_n$ be variables appearing in $\Gamma \longrightarrow \Delta$, and let $\varphi_1, \ldots, \varphi_n$ be formulas of size at most m. If we substitute everywhere φ_i for x_i then we get a valid proof of size at most sm.*

PROOF. All the rules of PK are closed under substitution. □

The preceding lemma shows that we can lift a proof of a sequent by attaching stuff 'below'. The next lemma shows that we can also lift a proof by attaching stuff 'above'; this corresponds to deep inference.

Definition 4.11. The double sequent $P \longleftrightarrow Q$ is the pair of sequents $P \longrightarrow Q$ and $Q \longrightarrow P$.

LEMMA 4.12. *Let $P \longrightarrow Q$ be a sequent of size m, and $\varphi(x)$ be a formula of size n in which the variable x appears only once (other variables may also appear). The double sequent $\varphi(x|P) \longleftrightarrow \varphi(x|Q)$ has a cut-free proof from the double sequent $P \longleftrightarrow Q$ of size $O(n(m+n))$.*

PROOF. The proof is by structural induction. If $\varphi = x$ then there is nothing to prove. If $\varphi = \neg\psi$, then $\varphi(P) \longleftrightarrow \varphi(Q)$ follows from $\psi(P) \longleftrightarrow \psi(Q)$ by four applications of the $\neg$ introduction rules.
If $\varphi = \psi \wedge \chi$, then assume, without loss of generality, that x appears in χ. We use the following proof twice:

$$\dfrac{\dfrac{\psi \longrightarrow \psi}{\psi \wedge \chi(P) \longrightarrow \psi}\,\wedge\mathsf{L} \qquad \dfrac{\chi(P) \longrightarrow \chi(Q)}{\psi \wedge \chi(P) \longrightarrow \chi(Q)}\,\wedge\mathsf{L}}{\psi \wedge \chi(P) \longrightarrow \psi \wedge \chi(Q)}\,\wedge\mathsf{R}$$

Each instance of the proof uses a different assumption. A similar proof works if the main connective is $\vee$ instead of $\wedge$. In total, there are $O(n)$ sequents of size $O(m+n)$. □

When the variable x appears several times in φ, we can prove a similar statement using the cut rule.

LEMMA 4.13. *Let $P \longrightarrow Q$ be a sequent of size m, φ be a formula of size n, and x be a variable. The sequent $\varphi(x|P) \longrightarrow \varphi(x|Q)$ has a proof of size $O(mn(m+n))$ from the sequent $P \longrightarrow Q$ cutting on formulas of depth d, where d is the maximal depth of a formula obtained from φ by replacing some of the occurrences of x by P, and others by Q.*

PROOF. Suppose that x occurs $\ell \leq m$ times in φ. Define hybrid formulas $\varphi_0 = \varphi(x|P), \ldots, \varphi_\ell = \varphi(x|Q)$ as follows: in the formula φ_t, the first t occurrences of x are replaced by Q, and the rest by P. For each $0 \leq t < \ell$, Lemma 4.12 shows how to prove $\varphi_t \longrightarrow \varphi_{t+1}$ from $P \longrightarrow Q$ in size $O(n(m+n))$. In total, these proofs take up size

$O(\ell n(m+n)) = O(mn(m+n))$. We can deduce $\varphi_0 \longrightarrow \varphi_\ell$ (i.e., $\varphi(x|P) \longrightarrow \varphi(x|Q)$) using $\ell \leq m$ applications of the cut rule. □

We next state two easy lemmas on dualization.

LEMMA 4.14. *Let φ be a formula of size n in negation normal form. The double sequents $\mathrm{M}(\varphi), \varphi \longleftrightarrow$ and $\mathrm{M}(\varphi) \longleftrightarrow \neg\varphi$ have a cut-free proof of size $O(n^2)$.*

PROOF. We construct inductively proofs of the double sequent $\mathrm{M}(\varphi), \varphi \longleftrightarrow$. From this, we conclude the double sequent $\mathrm{M}(\varphi) \longleftrightarrow \neg\varphi$ using two applications of the $\neg$ introduction rules.

If $\varphi = x$ or $\varphi = \neg x$ then required double sequent is proved as follows:

$$\dfrac{x \longrightarrow x}{\neg x, x \longrightarrow}\,\neg\mathsf{L} \qquad \dfrac{x \longrightarrow x}{\longrightarrow \neg x, x}\,\neg\mathsf{R}.$$

If $\varphi = \psi \wedge \chi$ then the proof is

$$\dfrac{\dfrac{\mathrm{M}(\psi), \psi \longrightarrow}{\mathrm{M}(\psi), \psi \wedge \chi \longrightarrow}\,\wedge\mathsf{L} \qquad \dfrac{\mathrm{M}(\chi), \chi \longrightarrow}{\mathrm{M}(\chi), \psi \wedge \chi \longrightarrow}\,\wedge\mathsf{L}}{\mathrm{M}(\psi) \vee \mathrm{M}(\chi), \psi \wedge \chi \longrightarrow}\,\vee\mathsf{L}$$

$$\dfrac{\dfrac{\longrightarrow \mathrm{M}(\psi), \psi}{\longrightarrow \mathrm{M}(\psi) \vee \mathrm{M}(\chi), \psi}\,\vee\mathsf{R} \qquad \dfrac{\longrightarrow \mathrm{M}(\chi), \chi}{\longrightarrow \mathrm{M}(\psi) \vee \mathrm{M}(\chi), \chi}\,\vee\mathsf{R}}{\longrightarrow \mathrm{M}(\psi) \vee \mathrm{M}(\chi), \psi \wedge \chi}\,\wedge\mathsf{R}$$

If $\varphi = \psi \vee \chi$ then the proof is similar. In all, we have $O(n)$ sequents of size $O(n)$. □

The second lemma allows us to lift an equivalence to its dualized version.

LEMMA 4.15. *Let φ, ψ be formulas in negation normal form. Suppose that the double sequent $\varphi \longleftrightarrow \psi$ has a proof of size s cutting on formulas of depth at most D. Then the double sequent $\mathrm{M}(\varphi) \longleftrightarrow \mathrm{M}(\psi)$ has a proof of size $O(s)$ cutting on formulas of depth at most $D + 1$.*

PROOF. For a formula A, denote by $\bar{A}$ the formula obtained from A by replacing each occurrence of a negative literal $\neg x$ by x, and each occurrence of a positive literal x (not occurring in the context $\neg x$) by $\neg x$. The depth of $\bar{A}$ is at most one more than the depth of A.

Given a proof Π, if we replace every formula A with $\bar{A}$, then most of the derivation steps remain valid. Indeed, the structural rules and the cut rules always remain valid, as are the logical rules other than the negation rules. The negation rules are valid unless $\overline{\neg A} \neq \neg\bar{A}$. This, in turn, only happens when $A = x$, where the invalid derivation in question is

$$\dfrac{\Gamma \longrightarrow \neg x, \Delta}{x, \Gamma \longrightarrow \Delta}\,\cancel{\neg L}$$

or its counterpart from the right. This derivation can be implemented as follows:

$$\dfrac{\dfrac{\Gamma \longrightarrow \neg x, \Delta}{\neg\neg x, \Gamma \longrightarrow \Delta}\,\neg\mathsf{L} \qquad \dfrac{}{x \longrightarrow \neg\neg x}\,\mathbf{A}\mathsf{x}, \neg\mathsf{L}, \neg\mathsf{R}}{x, \Gamma \longrightarrow \Delta}\,\mathbf{C}\mathsf{ut}$$

If we take the corresponding proof of $\bar{\varphi} \longrightarrow \bar{\psi}$, switch $\wedge$ with $\vee$, and switch the side of each formula in the proof, then we get a valid proof of $\mathrm{M}(\psi) \longrightarrow \mathrm{M}(\varphi)$: it is straightforward to check that all the rules remain valid under this transformation. Similarly we can obtain a proof of the other sequent. □

4.4. Moving down the depth vector

In this section we show how to prove the equivalence of two flattened forms of the same formula which correspond to two different depth vectors. As a preliminary step, we show how to prove that the two flattened forms $\mathrm{C}(\varphi; \vec{d})$ and $\mathrm{D}(\varphi; \vec{d})$ are equivalent using a recursive construction.

LEMMA 4.16. *Let φ be a formula, and $\vec{d} = d_1, \ldots, d_k$ be a vector of increasing positive integers. The double sequent $\mathrm{C}(\varphi; \vec{d}) \longleftrightarrow \mathrm{D}(\varphi; \vec{d})$ has a proof of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k + 4$, where $x = \mathrm{ex}(\varphi; \vec{d})$.*

PROOF. The proof is by induction on k. When $k = 0$, we prove that the CNF and DNF forms are equivalent using Lemma 4.9.

If $k > 0$, let ψ be the part of φ up to level d_1, and define $\vec{e} = d_2 - d_1, \ldots, d_k - d_1$. According to Definition 4.3, $\mathrm{C}(\varphi; \vec{d})$ is obtained from $\mathrm{CNF}(\psi)$ by replacing each positively occuring subformula literal v_χ by $\mathrm{D}(\chi; \vec{e})$, and each negatively occuring subformula literal $\neg v_\chi$ by $\mathrm{M}(\mathrm{C}(\chi; \vec{e}))$. Analogously, $\mathrm{D}(\varphi; \vec{d})$ is obtained from $\mathrm{DNF}(\psi)$ by replacing v_χ by $\mathrm{C}(\chi; \vec{e})$ and $\neg v_\chi$ by $\mathrm{M}(\mathrm{D}(\chi; \vec{e}))$. We also define a hybrid form $\mathrm{H}(\varphi; \vec{d})$ obtained from $\mathrm{CNF}(\psi)$ by replacing v_χ by $\mathrm{C}(\chi; \vec{e})$ and $\neg v_\chi$ by $\mathrm{M}(\mathrm{D}(\chi; \vec{e}))$.

For each subformula χ, the induction hypothesis gives us a proof of the double sequent $\mathrm{C}(\chi; \vec{e}) \longleftrightarrow \mathrm{D}(\chi; \vec{e})$ of size $2^{O((k-1)2^x)}$ cutting on formulas of depth at most $k + 3$, and Lemma 4.15 gives us a proof of $\mathrm{M}(\mathrm{C}(\chi; \vec{e})) \longleftrightarrow \mathrm{M}(\mathrm{D}(\chi; \vec{e}))$ of similar size cutting on formulas of depth at most $k + 4$. Since there are $2^{O(x)}$ subformulas, in total these proofs have size $2^{O(k2^x)}$. Several applications of Lemma 4.13 give us a proof of $\mathrm{C}(\varphi; \vec{d}) \longleftrightarrow \mathrm{H}(\varphi; \vec{d})$ of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k + 4$.

Lemma 4.9 gives us a proof of $\mathrm{CNF}(\psi) \longleftrightarrow \mathrm{DNF}(\psi)$ of size $2^{O(x)}$ which cuts only on variables. Lemma 4.10 lifts this to a proof of the double sequent $\mathrm{H}(\varphi; \vec{d}) \longleftrightarrow \mathrm{D}(\varphi; \vec{d})$ of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k + 2$. Cutting over $\mathrm{H}(\varphi; \vec{d})$ (a formula of depth $k + 4$), we obtain the desired proof of $\mathrm{C}(\varphi; \vec{d}) \longleftrightarrow \mathrm{D}(\varphi; \vec{d})$. □

The following lemmas show how to “split” a layer in a flattened form. We start with the special case in which it is the first layer which is split.

LEMMA 4.17. *Let φ be a formula, $\vec{d} = d_1, \ldots, d_k$ a vector of increasing positive integers, and $\delta < d_1$ be a positive integer. The double sequents $\mathrm{C}(\varphi; \vec{d}) \longleftrightarrow \mathrm{C}(\varphi; \delta, \vec{d})$ and $\mathrm{D}(\varphi; \vec{d}) \longleftrightarrow \mathrm{D}(\varphi; \delta, \vec{d})$ have proofs of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k + 4$, where $x = \mathrm{ex}(\varphi; \vec{d})$.*

PROOF. We show how to prove the first double sequent; the other one is proven in the same way.

Let ψ be the part of φ up to level d_1, and define $\vec{e} = d_2 - d_1, \ldots, d_k - d_1$. According to Definition 4.3, $\mathrm{C}(\varphi; \vec{d})$ is obtained from $\mathrm{CNF}(\psi)$ by replacing each positively occuring subformula literal v_χ by $\mathrm{D}(\chi; \vec{e})$, and each negatively occuring subformula literal $\neg v_\chi$ by $\mathrm{M}(\mathrm{C}(\chi; \vec{e}))$. Similarly, $\mathrm{C}(\varphi; \delta, \vec{d})$ is obtained from $\mathrm{C}(\psi; \delta)$ by replacing v_χ by $\mathrm{C}(\chi; \vec{e})$ and $\neg v_\chi$ by $\mathrm{M}(\mathrm{D}(\chi; \vec{e}))$. We also define a hybrid form $\mathrm{H}(\varphi; \vec{d})$ obtained from $\mathrm{CNF}(\psi)$ by replacing v_χ by $\mathrm{C}(\chi; \vec{e})$ and $\neg v_\chi$ by $\mathrm{M}(\mathrm{D}(\chi; \vec{e}))$.

For each subformula χ, Lemma 4.16 gives us a proof of $\mathrm{C}(\chi; \vec{e}) \longleftrightarrow \mathrm{D}(\chi; \vec{e})$ of size $2^{O((k-1)2^x)}$ cutting on formulas of depth at most $k+3$, and Lemma 4.15 gives us a proof of $\mathrm{M}(\mathrm{C}(\chi;\vec{e})) \longleftrightarrow \mathrm{M}(\mathrm{D}(\chi;\vec{e}))$ of similar size cutting on formulas of depth at most $k+4$. Since there are $2^{O(x)}$ subformulas, in total these proofs have size $2^{O(k2^x)}$. Several applications of Lemma 4.13 give us a proof of $\mathrm{C}(\varphi;\vec{d}) \longleftrightarrow \mathrm{H}(\varphi;\vec{d})$ of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k+4$.

Lemma 4.9 gives us a proof of $\mathrm{CNF}(\psi) \longleftrightarrow \mathrm{C}(\psi;\delta)$ of size $2^{O(x)}$ which cuts only on variables. Lemma 4.10 lifts this to a proof of the double sequent $\mathrm{H}(\varphi;\vec{d}) \longleftrightarrow \mathrm{C}(\varphi;\delta,\vec{d})$ of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k+2$. Cutting over $\mathrm{H}(\varphi;\vec{d})$ (a formula of depth $k+4$), we obtain the desired proof of $\mathrm{C}(\varphi;\vec{d}) \longleftrightarrow \mathrm{C}(\varphi;\delta,\vec{d})$. □

The general case is given by the following lemma.

LEMMA 4.18. *Let φ be a formula, $\vec{d} = d_1,\dots,d_k$ a vector of increasing positive integers, and $d_i < \delta < d_{i+1}$, where $1 \le i < k$. Define $\vec{e} = d_1,\dots,d_i,\delta,d_{i+1},\dots,d_k$. The double sequents $\mathrm{C}(\varphi;\vec{d}) \longleftrightarrow \mathrm{C}(\varphi;\vec{e})$ and $\mathrm{D}(\varphi;\vec{d}) \longleftrightarrow \mathrm{D}(\varphi;\vec{e})$ have proofs of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k+4$, where $x = \mathrm{ex}(\varphi;\vec{d})$.*

PROOF. We show how to prove the first double sequent; the other one is proven in the same way.

Assume for simplicity that i is even; the proof when i is odd is very similar. Let ψ be the portion of φ up to level d_i, and define $\vec{f} = d_1,\dots,d_i$ and $\vec{g} = d_{i+1}-d_i,\dots,d_k-d_i$. According to Definition 4.3, $\mathrm{C}(\varphi;\vec{d})$ is obtained from $\mathrm{C}(\psi;\vec{f})$ by replacing each positively occuring subformula literal v_χ by $\mathrm{C}(\chi;\vec{g})$, and each negatively occuring subformula literal $\neg v_\chi$ by $\mathrm{M}(\mathrm{D}(\chi;\vec{g}))$. Similarly, $\mathrm{C}(\varphi;\delta,\vec{d})$ is obtained from $\mathrm{C}(\psi;\vec{f})$ by replacing v_χ by $\mathrm{C}(\chi;\delta-d_i,\vec{g})$ and $\neg v_\chi$ by $\mathrm{M}(\mathrm{D}(\chi;\delta-d_i,\vec{g}))$.

For each subformula χ, Lemma 4.17 gives us proofs of $\mathrm{C}(\chi;\vec{g}) \longleftrightarrow \mathrm{C}(\chi;\delta-d_i,\vec{g})$ and $\mathrm{D}(\chi;\vec{g}) \longleftrightarrow \mathrm{D}(\chi;\delta-d_i,\vec{g})$ of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k+3$, and Lemma 4.15 gives us a proof of $\mathrm{M}(\mathrm{D}(\chi;\vec{g})) \longleftrightarrow \mathrm{M}(\mathrm{D}(\chi;\delta-d_i,\vec{g}))$ of similar size cutting on formulas of depth at most $k+4$. Several applications of Lemma 4.13 give us the desired proof of $\mathrm{C}(\varphi;\vec{d}) \longleftrightarrow \mathrm{C}(\varphi;\vec{e})$; these proofs have size $2^{O(k2^x)}$ and cut on formulas of depth at most $k+4$. □

In order to be able to "shift" the depth vector down, we need the individual depths to be not only increasing, but in fact increasing by at least 2.

Definition 4.19. A vector $\vec{d} = d_1,\dots,d_k$ is *strongly increasing* if $d_1 \ge 1$ and $d_{i+1} \ge d_i + 2$ for each $1 \le i < k$.

We are ready to prove the fundamental lemma allowing us to shift the depth vector.

LEMMA 4.20. *Let φ be a formula, and $\vec{d} = d_1,\dots,d_k$ be a vector of strongly increasing positive integers. Define $\vec{e} = 1, d_1+1,\dots,d_k+1$. The double sequents $\mathrm{C}(\varphi;\vec{d}) \longleftrightarrow \mathrm{C}(\varphi;\vec{e})$ and $\mathrm{D}(\varphi;\vec{d}) \longleftrightarrow \mathrm{D}(\varphi;\vec{e})$ have proofs of size $2^{O(k2^x)}$ cutting on formulas of depth at most $k+4$, where $x = \max(\mathrm{ex}(\varphi;\vec{d}),\mathrm{ex}(\varphi;\vec{e}))$.*

PROOF. We show how to prove the first double sequent; the other one is proven in the same way.

Lemma 4.17 and Lemma 4.18 show how to prove the equivalence of two flattened forms, where the second one has one extra layer beyond the first one. In order to 'move' d_1 to d_1+1, we prove the following double sequents:

$$\mathrm{C}(\varphi;d_1,d_2,\dots,d_k) \longleftrightarrow \mathrm{C}(\varphi;d_1,d_1+1,d_2,\dots,d_k),$$
$$\mathrm{C}(\varphi;d_1,d_1+1,d_2,\dots,d_k) \longleftrightarrow \mathrm{C}(\varphi;d_1+1,d_2,\dots,d_k).$$

Continuing in the same way, we can 'migrate' $\vec{d}$ to $\vec{e}$, adding the extra $e_1 = 1$ at the end using Lemma 4.17. Each flattened form in the interim has depth at most $k+4$. By cutting all the intermediate flattened forms, we obtain the desired double sequent. □

4.5. Putting it together

In this section we show how to transform a Frege proof to an AC^0-Frege proof. We begin by showing that the C and D operators respect the logical connectives.

LEMMA 4.21. *Let φ, ψ be formulas, and $\vec{d}$ be a vector of strongly increasing positive integers of length k. The double sequents*

$$\mathrm{C}(\varphi \land \psi; \vec{d}) \longleftrightarrow \mathrm{C}(\varphi; \vec{d}) \land \mathrm{C}(\psi; \vec{d}), \qquad \mathrm{C}(\varphi \lor \psi; \vec{d}) \longleftrightarrow \mathrm{C}(\varphi; \vec{d}) \lor \mathrm{C}(\psi; \vec{d}),$$
$$\mathrm{D}(\varphi \land \psi; \vec{d}) \longleftrightarrow \mathrm{D}(\varphi; \vec{d}) \land \mathrm{D}(\psi; \vec{d}), \qquad \mathrm{D}(\varphi \lor \psi; \vec{d}) \longleftrightarrow \mathrm{D}(\varphi; \vec{d}) \lor \mathrm{D}(\psi; \vec{d}),$$

have proofs of size $2^{O(k2^x)}$ with cuts on formulas of depth at most $k+4$, where $x = \mathrm{ex}(\varphi \land \psi; \vec{d})$.

PROOF. We show how to prove the first double sequent; the rest are proven in the same way.

Let $\vec{e} = 1, d_1 + 1, \ldots, d_k + 1$ be the vector defined in Lemma 4.20. We calculate $\mathrm{D}(\varphi \land \psi; \vec{e})$. Using the recipe of Definition 4.3, we first calculate $\mathrm{DNF}(v_\varphi \land v_\psi) = v_\varphi \land v_\psi$. Into this DNF we substitute $v_\varphi = \mathrm{C}(\varphi; \vec{d})$ and $v_\psi = \mathrm{C}(\psi; \vec{d})$. Therefore $\mathrm{D}(\varphi \land \psi; \vec{e}) = \mathrm{C}(\varphi; \vec{d}) \land \mathrm{C}(\psi; \vec{d})$. The proof now becomes obvious, along the following lines.

Using Lemma 4.16, we prove $\mathrm{C}(\varphi \land \psi; \vec{d}) \longleftrightarrow \mathrm{D}(\varphi \land \psi; \vec{d})$. Using Lemma 4.20, we prove $\mathrm{D}(\varphi \land \psi; \vec{d}) \longleftrightarrow \mathrm{C}(\varphi; \vec{d}) \land \mathrm{C}(\psi; \vec{d})$. The proof is completed by an application of the cut rule. □

LEMMA 4.22. *Let φ be a formula, and $\vec{d}$ be a vector of strongly increasing positive integers of length k. The double sequents $\mathrm{C}(\neg\varphi; \vec{d}) \longleftrightarrow \neg\,\mathrm{C}(\varphi; \vec{d})$ and $\mathrm{D}(\neg\varphi; \vec{d}) \longleftrightarrow \neg\,\mathrm{D}(\varphi; \vec{d})$ have proofs of size $2^{O(k2^x)}$ with cuts on formulas of depth at most $k+4$, where $x = \mathrm{ex}(\neg\varphi; \vec{d})$.*

PROOF. We show how to prove the first double sequent; the other one is proven in the same way.

Let $\vec{e} = 1, d_1{+}1, \ldots, d_k{+}1$ be the vector defined in Lemma 4.20. We calculate $\mathrm{C}(\neg\varphi; \vec{e})$. Using the recipe of Definition 4.3, we first calculate $\mathrm{CNF}(\neg v_\varphi) = \neg v_\varphi$. Into this CNF we substitute $\neg v_\varphi = \mathrm{M}(\mathrm{C}(\varphi; \vec{d}))$. Therefore $\mathrm{C}(\neg\varphi; \vec{e}) = \mathrm{M}(\mathrm{C}(\varphi; \vec{d}))$. The proof now becomes obvious, along the following lines.

Lemma 4.20 shows how to prove $\mathrm{C}(\neg\varphi; \vec{d}) \longleftrightarrow \mathrm{M}(\mathrm{C}(\varphi; \vec{d}))$. Lemma 4.14 shows how to prove $\mathrm{M}(\mathrm{C}(\varphi; \vec{d})) \longleftrightarrow \neg\,\mathrm{C}(\varphi; \vec{d})$. The proof is completed by applying the cut rule. □

The preceding lemmas allow us to unroll flattened forms.

LEMMA 4.23. *Let φ be a formula, and $\vec{d}$ be a vector of strongly increasing positive integers of length k. The double sequents $\varphi \longleftrightarrow \mathrm{C}(\varphi; \vec{d})$ and $\varphi \longleftrightarrow \mathrm{D}(\varphi; \vec{d})$ have proofs of size $2^{O(k2^x)}$ with cuts on formulas of depth at most $k+4$, where $x = \mathrm{ex}(\varphi; \vec{d})$.*

PROOF. We show how to prove the first double sequent; the other one is proven in the same way.

The proof is by structural induction. If φ is a literal then there is nothing to prove. If $\varphi = \neg\psi$, then use Lemma 4.22 to prove $\mathrm{C}(\varphi; \vec{d}) \longleftrightarrow \neg\,\mathrm{C}(\psi; \vec{d})$. The induction hypothesis gives us a proof of $\psi \longleftrightarrow \mathrm{C}(\psi; \vec{d})$; move both ψ and its flattened form to the other side using four $\neg$ introduction rules, and apply cut twice to prove the required double sequent.

If $\varphi = \psi \wedge \chi$ then start with proofs of the following sequents, obtained by Lemma 4.21 and the induction hypothesis:

$$\mathrm{C}(\varphi;\vec{d}) \longleftrightarrow \mathrm{C}(\psi;\vec{d}) \wedge \mathrm{C}(\chi;\vec{d}), \qquad \psi \longleftrightarrow \mathrm{C}(\psi;\vec{d}), \qquad \chi \longleftrightarrow \mathrm{C}(\chi;\vec{d}).$$

Now prove $\mathrm{C}(\psi;\vec{d}) \wedge \mathrm{C}(\chi;\vec{d}) \longleftrightarrow \psi \wedge \chi$ as follows:

$$\dfrac{\dfrac{\mathrm{C}(\psi;\vec{d}) \longrightarrow \psi}{\mathrm{C}(\psi;\vec{d}) \wedge \mathrm{C}(\chi;\vec{d}) \longrightarrow \psi}\wedge\mathsf{L} \qquad \dfrac{\mathrm{C}(\chi;\vec{d}) \longrightarrow \chi}{\mathrm{C}(\psi;\vec{d}) \wedge \mathrm{C}(\chi;\vec{d}) \longrightarrow \chi}\wedge\mathsf{L}}{\mathrm{C}(\psi;\vec{d}) \wedge \mathrm{C}(\chi;\vec{d}) \longrightarrow \psi \wedge \chi}\wedge\mathsf{R}$$

The other sequent is proved similarly. Complete the proof using the cut rule. The case $\varphi = \psi \vee \chi$ is similar. □

The proof of the main theorem is now simple.

LEMMA 4.24. *Let φ be a formula provable in Frege in size s using a proof with maximum logical depth D. For every k there is an AC^0_{k+4}-Frege proof of φ of size $s2^{O(k2^{D/k})}$.*

PROOF. Let $\vec{d} = 2\lceil D/(2k)\rceil, 4\lceil D/(2k)\rceil, \ldots, 2(k-1)\lceil D/(2k)\rceil$, and note that $\vec{d}$ is strongly increasing. The extent of each formula with respect to $\vec{d}$ is at most $x = 2\lceil D/(2k)\rceil$. Take the original proof and replace each formula ψ by $\mathrm{C}(\psi;\vec{d})$. Each application of a rule is still valid, but the proof as a whole isn't valid since not all formulas are in flattened form. We address this issue by tampering with the introduction rules, as in the following example, corresponding to the right $\wedge$ introduction rule:

$$\dfrac{\dfrac{\Gamma \longrightarrow \Delta, \mathrm{C}(\psi;\vec{d}) \qquad \Gamma \longrightarrow \Delta, \mathrm{C}(\chi;\vec{d})}{\Gamma \longrightarrow \Delta, \mathrm{C}(\psi;\vec{d}) \wedge \mathrm{C}(\chi;\vec{d})}\wedge\mathsf{R} \qquad \dfrac{}{\mathrm{C}(\psi;\vec{d}) \wedge \mathrm{C}(\chi;\vec{d}) \longrightarrow \mathrm{C}(\psi \wedge \chi;\vec{d})}\,\text{Lem. 4.21}}{\Gamma \longrightarrow \Delta, \mathrm{C}(\psi \wedge \chi;\vec{d})}\,\text{Cut}$$

Applying the same transformation for all introduction rules, we are left with a valid proof of $\longrightarrow \mathrm{C}(\varphi;\vec{d})$, where each sequent is now replaced by sequents of total size $2^{O(k2^x)}$; the total size so far is $s2^{O(k2^x)}$. Lemma 4.23 proves $\mathrm{C}(\varphi;\vec{d}) \longrightarrow \varphi$, and the proof is complete by cutting on $\mathrm{C}(\varphi;\vec{d})$.

The lemmas we used employ cuts of depth at most $k+4$. All cuts in the original proof now cut flattened formulas, which are of depth at most $k+3$. □

PROOF OF THEOREM 3.1. Reckhow's Theorem (Theorem 2.1) supplies us with an $\mathrm{AC}^0_{O(\log s)}$ proof of φ of size $s^{O(1)}$. The theorem now follows by substituting $D = C\log(s)$ in Lemma 4.24. □

REFERENCES

Michael Alekhnovich and Alexander Razborov. 2001. Resolution is Not Automatizable Unless W[P] is Tractable. In *Proceedings of the Fourty-Second Annual Symposium on Foundations of Compuer Science*. The Institute of Electrical and Electronics Engineers (IEEE), Piscataway, NJ, 210–219.

Eric Allender, Lisa Hellerstein, Paul McCabe, Toniann Pitassi, and Michael Saks. 2008. Minimizing Disjunctive Normal Form Formulas and AC^0 Circuits Given a Truth Table. *SIAM J. Comput.* 38, 1 (2008), 63–84.

Albert Atserias and Elitza Maneva. 2011. Mean-payoff games and propositional proofs. *Information and Computation* 209, 4 (2011), 664–691.

Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thom. 2013. A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. Cryptology ePrint Archive, Report 2013/400. (2013). http://eprint.iacr.org/.

Paul W. Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan Woods. 1992. Exponential Lower Bounds for the Pigeonhole Principle. In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Computing*. Association for Computing Machinery (ACM), New York, NY, 200–220.

Maria Luisa Bonet, Samuel R. Buss, and Toniann Pitassi. 1995. Are there Hard Examples for Frege Systems? In *Feasible Mathematics II*. Birkhäuser, Boston, 30–56.

Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. 2004. Non-Automatizability of Bounded-Depth Frege Proofs. *Computational Complexity* 13, 1-2 (2004), 47–68.

Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. 2000. On Interpolation and Automatization for Frege Systems. *SIAM J. Comput.* 29, 6 (2000), 1939–1967.

Richard P. Brent. 1974. The parallel evaluation of general arithmetic expressions. *J. ACM* 21 (1974), 201–206.

Samuel R. Buss. 1987. Polynomial size proofs of the pigeonhole principle. *Journal of Symbolic Logic* 57 (1987), 916–927.

Stephen A. Cook and Robert A. Reckhow. 1979. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic* 44, 1 (1979), 36–50.

Xudong Fu and Alasdair Urquhart. 1996. Simplified Lower Bounds for Propositional Proofs. *Notre Dame Journal of Formal Logic* 37, 4 (1996), 523–544.

Kaveh Ghasemloo and Stephen A. Cook. 2013. Theories for Subexponential-size Bounded-depth Frege Proofs. In *Computer Science Logic 2013 (CSL 2013) (Leibniz International Proceedings in Informatics (LIPIcs))*, Simona Ronchi Della Rocca (Ed.), Vol. 23. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 296–315. DOI:http://dx.doi.org/10.4230/LIPIcs.CSL.2013.296

Johan Håstad. 1987. *Computational limitations of small depth circuits*. Ph.D. Dissertation. Massachusetts Institute of Technology.

Armin Haken. 1985. The intractability of resolution. *Theoretical Computer Science* 39 (1985), 297–305.

Antoine Joux. 2013. A new index calculus algorithm with complexity $L(1/4 + o(1))$ in very small characteristic. Cryptology ePrint Archive, Report 2013/095. (2013). http://eprint.iacr.org/.

Jan Krajíček. 1994. Lower bounds to the size of constant-depth propositional proofs. *Journal of Symbolic Logic* 59, 1 (March 1994), 73–86.

Jan Krajíček. 1995. *Bounded arithmetic, propositional logic, and complexity theory*. Cambridge University Press, New York, NY, USA.

Jan Krajíček. 1997. Interpolation Theorems, Lower Bounds for Proof Systems, and Independence Results for Bounded Arithmetic. *J. Symbolic Logic* 62 (1997), 457–486. Issue 2.

Jan Krajíček. 2011. *Forcing with random variables and proof complexity*. London Mathematical Society, London, United Kingdom.

Jan Krajíček, Pavel Pudlák, and Alan Woods. 1995. An exponential lower bound to the size of bounded depth Frege proofs of the Pigeonhole Principle. *Random Structures and Algorithms* 7, 1 (Aug. 1995), 15–39. DOI:http://dx.doi.org/10.1002/rsa.3240070103

Leonid A. Levin. 1973. Universal sequential search problems. *Problems of Information Transmission (translated from Russian)* 9 (1973), 115–116. Issue 3.

Sebastian Müller. 2013. Polylogarithmic cuts in models of V^0. *Logical Methods in Computer Science* 9, 1 (2013), 16.

V. A. Nepomnjaščij. 1970. Rudimentary predicates and Turing calculations. *Soviet Mathematics Doklady* 11 (1970), 1462–1465.

Knot Pipatsrisawat and Adnan Darwiche. 2011. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.* 175, 2 (2011), 512–525.

Toniann Pitassi, Paul Beame, and Russell Impagliazzo. 1993. Exponential lower bounds for the pigeonhole principle. *Computational Complexity* 3, 2 (1993), 97–140. DOI:http://dx.doi.org/10.1007/BF01200117

Robert A. Reckhow. 1976. *On the Lengths of Proofs in the Propositional Calculus*. Ph.D. Dissertation. University of Toronto.

Michael Sipser. 1983. Borel Sets and Circuit Complexity. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing (STOC '83)*. ACM, New York, NY, USA, 61–69. DOI:http://dx.doi.org/10.1145/800061.808733

P. M. Spira. 1971. On time-hardware complexity tradeoffs for Boolean functions. In *Proc. 4th Hawaii Symp. on System Sciences*. The Institute of Electrical and Electronics Engineers (IEEE), Piscataway, NJ, 525–527.

Received Month Year; revised Month Year; accepted Month Year