Electronic Colloquium on Computational Complexity, Report No. 199 (2023)

Towards P $\neq$ NP from Extended Frege lower bounds

Ján Pich
University of Oxford

Rahul Santhanam
University of Oxford

December 9, 2023

Abstract

We give a new approach to the fundamental question of whether proof complexity lower bounds for concrete propositional proof systems imply super-polynomial Boolean circuit lower bounds.

For any poly-time computable function f, we define the witnessing formulas $w_n^k(f)$, which are propositional formulas stating that for any circuit C of size n^k on n variables and for any formula ϕ of size n, either C computes a satisfying assignment to ϕ or f verifiably refutes that C computes SAT on instances of length n. We show that if the witnessing formulas are tautologies, then any super-polynomial lower bound for Extended Frege augmented with $w_n^k(f)$ axioms implies that SAT requires super-polynomial size Boolean circuits. We also give an unconditional equivalence between proof complexity lower bounds for a concretely defined strong (non-uniform) propositional proof system and super-polynomial circuit lower bounds for the Discrete Logarithm problem.

We give consequences for the meta-mathematics of several major questions in computational complexity, including whether one-way functions can be based on the worst-case hardness of NP, whether there is a dichotomy between one-way functions and worst-case learning with membership queries over the uniform distribution, and whether there are feasibly constructible anti-checkers for Satisfiability. We show that for each of these questions, provability of a positive answer in *any* system of bounded arithmetic would imply new connections between propositional proof complexity and circuit complexity.

Our results rely on a new notion of "self-provability" of upper bounds, which might be independently interesting.

1 Introduction

Proof complexity studies the lengths of proofs of tautologies in propositional proof systems. One of the motivations for studying proof complexity is that the P versus NP

ISSN 1433-8092

problem can be approached by showing proof complexity lower bounds. Cook and Reckhow [22] showed that $\mathsf{NP} \neq \mathsf{coNP}$ iff every propositional proof system has a hard sequence of tautologies that require superpolynomial proof size. The Cook-Reckhow program [6, 38] proceeds by proving lower bounds on proof size for increasingly powerful proof systems. One issue with this program is that it is unclear if there is a *concrete* proof system such that lower bounds for that system would imply $\mathsf{P} \neq \mathsf{NP}$, as the separation is only known to follow if we have lower bounds for *every* proof system, including very powerful proof systems that are not studied in practice. In contrast, the circuit complexity approach to the P versus NP problem only requires super-polynomial lower bounds for a concrete circuit model, namely general Boolean circuits.

In this paper, we present a new approach to the question of whether superpolynomial lower bounds for concrete proof systems have implications for longstanding open problems in complexity theory such as P versus NP. Our approach is based on the new concept of *self-provability* of Boolean formulas, and on the use of *feasible witnessing of lower bounds* to establish self-provability. In the first part of our paper, we use this approach to connect proof complexity lower bounds for propositional proof systems to strong computational complexity lower bounds such as $\mathsf{P} \neq \mathsf{NP}$. In the second part of our paper, we derive some intriguing *meta-mathematical* consequences of our results in the first part. We show that *any* approach to certain foundational questions in cryptography and learning, such as the question of whether one-way functions can be based on $\mathsf{P} \neq \mathsf{NP}$, or whether there is a dichotomy between cryptographic pseudo-randomness and efficient learning, is also an approach to deriving strong Boolean complexity lower bounds from proof complexity lower bounds!

In the rest of this introduction, we first discuss the question of connecting proof complexity and circuit complexity. We then present our approach and the connections we derive from it. In the final subsection, we present the meta-mathematical implications of our approach in detail.

1.1 Proof Complexity vs Circuit Complexity

On the surface, there are differences between proof complexity and circuit complexity. In proof complexity, we are interested in the sizes of proofs where every line is a tautology, and hence it is unclear how computational complexity could be relevant. Also, we know that strong circuit lower bounds hold for *most* Boolean functions by a simple counting argument, while super-polynomial proof complexity lower bounds for *any* sequence of tautologies, whether explicit or not, is unknown for strong proof systems.

Despite this, proof complexity and circuit complexity are closely linked together, especially for weak proof systems. Lower bound techniques for proof systems such as Resolution [26] and AC^0-Frege [4, 14, 40] are closely related to circuit complexity lower bound techniques for circuit models such as monotone circuits and constant-depth circuits. More formally, there are "feasible interpolation" results for weak proof systems such as Resolution and Cutting Planes which allow us to derive proof complexity lower bounds from circuit lower bounds [35]. In the other direction, there has been much progress on "lifting" theorems [23], which yield monotone circuit lower bounds from lower bounds for Resolution and monotone span program lower bounds from lower bounds for the Nullstellensatz proof system. Such connections are thus far confined to weak proof systems, and there is cryptographic evidence that stronger proof systems - AC^0-Frege and above - do not have feasible interpolation [12, 11].

There are also known connections between algebraic circuit complexity and proof complexity [49]. Grochow and Pitassi [24] defined the Ideal Proof System (IPS): an algebraic proof system where the verification of proofs can be done by an efficient randomized algorithm for Polynomial Identity Testing. They showed that any super-polynomial lower bounds on IPS-proofs of propositional tautologies would imply $\mathsf{VNP} \neq \mathsf{VP}$ - the major open problem in algebraic circuit complexity. There has been follow-up work on variants and subsystems of IPS, including a version involving non-commutative algebraic formulas which is closely related to the Frege proof system [41]. Recently, an equivalence between $\mathsf{VNP} \neq \mathsf{VP}$ and IPS lower bounds for a certain explicit sequence of formulas was shown in [53]. There are also strong connections known between lower bounds for QBF proof systems and circuit lower bounds [7, 8].

However this still leaves open the question of formal connections between proof complexity lower bounds for strong propositional proof systems such as Extended Frege system EF and Boolean circuit lower bounds. No implications are known in either direction. Razborov [51] has proposed a set of conjectures connecting average-case circuit lower bounds to proof complexity lower bounds for Frege and EF, but we seem very far from establishing these conjectures.

We first observe that an implication from super-polynomial proof complexity lower bounds for *any* concrete propositional proof system to strong Boolean circuit lower bounds would yield unconditional circuit lower bounds we don't yet know how to prove.

Proposition 1. *Suppose that Q is a propositional proof system such that super-polynomial lower bounds on the Q-proof size for any sequence of tautologies implies that* $\mathsf{NP} \not\subseteq \mathsf{P/poly}$. *Then we have unconditionally that* $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

Proof. We consider two cases. The first is that there are polynomial-size Q-proofs for any sequence of tautologies. In this case $\mathsf{NP} = \mathsf{coNP}$, and hence the Polynomial Hierarchy collapses to NP. Suppose for the sake of contradiction that $\mathsf{NEXP} \subseteq \mathsf{P/poly}$. Then, by the main result of [29], $\mathsf{NEXP} = \mathsf{MA}$. Since MA is in the Polynomial Hierarchy, we have that $\mathsf{MA} = \mathsf{NP}$, and hence that $\mathsf{NEXP} = \mathsf{NP}$. But this contradicts the hierarchy theorem for non-deterministic time, and hence we must have that $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

The second case is that there are super-polynomial lower bounds on the Q-proof size for some sequence of tautologies. By the assumption in Proposition 1, we have that $\mathsf{NP} \not\subseteq \mathsf{P/poly}$, and hence that $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$. □

Since the NEXP vs $\mathsf{P/poly}$ problem seems out of reach of current lower bound techniques, it seems unlikely that we will be able to prove an unconditional connection from super-polynomial lower bounds for a propositional proof system to strong circuit lower bounds any time soon. However, there are still several possibilities for establishing connections. First, we can still hope to establish a connection under the assumption that $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$, or similar assumptions for exponential-time classes, which seem like much weaker lower bound assumptions than super-polynomial circuit lower bounds for NP. Second, we can hope to establish a connection unconditionally from super-polynomial lower bounds for a proof system with randomized or non-uniform verification - the proof of Proposition 1 needs that proofs are deterministically verifiable. Third, we can hope to establish a connection with a weaker conclusion, e.g., that $\mathsf{NP} \neq \mathsf{P}$, which is not ruled out by the argument above. Fourth, we can hope to establish a connection that doesn't work for any sequence of tautologies but only for certain specific sequences.

In this paper, we explore all four of these possibilities, and show several results that make progress towards realizing these possibilities. The new idea that is used in all of these results is the idea of *self-provability* of lower bounds, which we explain next.

1.2 Self-Provability and Witnessing

Is it possible that $\mathsf{NP} = \mathsf{P}$ but this fact is not provable in any standard theory, such as ZFC (Zermelo-Fraenkel Set Theory with Choice)? There have been investigations into the possibility that NP vs P is *independent* of ZFC [1], which if true would imply a positive answer to the question above. One might speculate that if $\mathsf{NP} = \mathsf{P}$ or similar upper bounds hold, there should also be efficient *proofs* of such upper bounds. In cases where this intuition is true, we say that the upper bound is *self-provable*. Self-provability in theories of bounded arithmetic is strongly relevant to the research direction of proving *consistency* of complexity lower bounds with a theory T. Indeed, if a complexity upper bound were self-provable in T, then the consistency of its negation with T would immediately imply that the negation holds unconditionally.

We study self-provability for circuit upper bounds in the propositional setting. Let S be an NP search problem on inputs of size n, and C a circuit of size $poly(n)$. We can formalize that C solves S on inputs of size n using a propositional statement $\phi_{C,n}$, which asserts that for each x of length n, either S has no solutions on x, or C outputs a solution to S. The question we pose is: if $\phi_{C,n}$ is a tautology, i.e., S is indeed solved by the polynomial-size circuit C on inputs of length n, does this imply that $\phi_{C,n}$ has short proofs in some concrete proof system Q?

Our goal is to show self-provability for interesting NP search problems S, either unconditionally or under reasonable assumptions. Note that if we show such a self-provability for some proof system Q, we immediately get an implication from proof complexity lower bounds to strong Boolean circuit lower bounds: if $\phi_{C,n}$ requires super-polynomial size

Q-proofs for each $poly(n)$-size circuit C, then S does not have poly-size circuits. Indeed, it's easy to see that this is an *equivalence* - the converse follows from soundness of the proof system Q.

The approach we use to show self-provability is *feasible witnessing*. We say that there is feasible witnessing of circuit lower bounds for a NP search problem S if there is an efficiently computable function f such that, given any circuit C that does not solve S correctly on inputs of length n, f produces a "counter-example" x together with a string z such that z is a solution to S on x and moreover C does not produce a valid solution on input x [1]. Note that we can assume without loss of generality that C only fails on instances of S that have valid solutions, as we can easily verify whether the output of C is a solution to S and hence enforce that C outputs $\perp$ on instances of S that do not have solutions. Intuitively, what we ask from a feasible witnessing procedure is a counter-example to the claim that C solves S, together with a way to verify the counter-example.

Feasible witnessing is related to several other notions that have been studied in bounded arithmetic and in computational complexity. Systems of bounded arithmetic such as Cook's theory PV_1 and Buss's theory S_2^1 enjoy witnessing theorems [13, 38] for provable statements of low quantifier complexity. For example, a proof of $\forall x \exists y R(x, y)$ in S_2^1 for a polynomial-time computable relation R yields an efficiently computable witness y for each x. Such witnessing theorems can be used to show unprovability results in the theory using complexity hardness assumptions. In contrast, we aim to reverse the implication and derive *provability* results for propositional proof systems from complexity easiness assumptions. Witnessing of complexity lower bounds has also been considered in purely computational contexts, through the notion of a "refuter" [33, 25, 5, 10, 17].

How do we use feasible witnessing to show self-provability for a search problem S? We argue by contradiction. Assume that the circuit C fails to solve S. Then we can run the feasible witnessing procedure f on C to produce a pair of strings x and z. By the correctness of the feasible witnessing procedure, we have that z is a solution to S on x and also that C does not produce a valid solution on input x. Note that both of these conditions can be checked efficiently, hence if either of them does not hold, we know that our original assumption that C fails to solve S must be flawed. This argument can be used to get short proofs in a sufficiently strong proof system of circuit upper bounds for S. Indeed, suppose that the correctness of the procedure f is provable in some theory T of bounded arithmetic. Then we show that if the propositional translation P_T of T simulates EF, we get short P_T-proofs of $\phi_{C,n}$ by translating the correctness proof of f and simulating the verification process for the failure of the feasible witnessing on C.

Interestingly, we reason using bounded arithmetic to get self-provability for propositional proof systems, but it is unclear how to get self-provability for theories of bounded

[1] We also need the correctness of f to be provable efficiently in the propositional proof system Q in which we show self-provability.

arithmetic.[2]

As a first application, we consider the Discrete Logarithm problem, which is widely believed to be hard for polynomial-size circuits. By using the random self-reducibility of Discrete Logarithm, together with a suitable derandomization assumption, we get feasible witnessing for the problem, and hence self-provability. This yields the following result. In the statement of the result below, by a "non-uniform" propositional proof system, we mean a sound and complete proof system where the verification of proofs can be done by polynomial-size circuits.

Theorem 1. *(Equivalences between Proof Complexity Lower Bounds and Circuit Lower Bounds for Discrete Logarithm)*

1. *Suppose there is a Boolean function $f \in \mathsf{E}$ such that f requires circuit size $2^{\Omega(n)}$ on average over the uniform distribution over inputs of length n, for all large enough n. Then there is a strong propositional proof system Q (simulating EF) and for each $k, n \in \mathbb{N}$ a set of formulas $F_{k,n}$ where each formula in $F_{k,n}$ is of size $n^{O(k)}$, such that the Discrete Logarithm problem for multiplicative groups $\mathbb{Z}_q^\times$ of integers modulo a prime $q \in (2^{n-1}, 2^n]$ does not have polynomial size circuits iff for each $k \in \mathbb{N}$ the sequence of (sets of) formulas $\{F_{k,n}\}_n$ does not have polynomial size Q-proofs.*[3]

2. *There is a concrete strong non-uniform propositional proof system Q (simulating EF) and for each $k, n \in \mathbb{N}$ a set of formulas $\{F_{k,n}\}$, where each set $F_{k,n}$ consists of formulas of size $n^{O(k)}$, such that the Discrete Logarithm problem for multiplicative groups $\mathbb{Z}_q^\times$ of integers modulo a prime $q \in (2^{n-1}, 2^n]$ does not have polynomial size circuits iff for each $k \in \mathbb{N}$ the sequence of (sets of) formulas $\{F_{k,n}\}_n$ does not have polynomial size Q-proofs.*

The first item of Theorem 1 gives an equivalence between propositional proof complexity lower bounds for a strong propositional proof system and strong circuit lower bounds for Discrete Logarithm, but conditional on the assumption that E requires large circuits (which is a complexity-theoretic assumption that is intuitively much weaker than the super-polynomial hardness of Discrete Logarithm). The second item gives an unconditional equivalence, but for a concrete strong non-uniform propositional proof system. [53] earlier give an unconditional equivalence between algebraic circuit lower bounds and proof complexity lower bounds for a non-uniform (in fact, randomized) proof system.

Indeed, the two items of Theorem 1 have essentially the same proof, involving consideration of a concrete proof system Q which is defined as EF with added axioms including a

[2] We could also eliminate the use of bounded arithmetic completely and work purely with propositional proof systems, but this would make our arguments far more lengthy and cumbersome.

[3] We use the natural convention that a sequence $\{F_n\}_n$ of sets of formulas does not have polynomial size proofs if there is a sequence of formulas $\{f_n\}_n$, where $f_n \in F_n$ for each $n \in \mathbb{N}$, such that $\{f_n\}_n$ does not have polynomial size proofs.

sequence of truth-table tautologies asserting that Boolean functions f_n are exponentially hard on average for Boolean circuits. The only reason that Item 1 is conditional is that we need lower bounds for E to be able to recognise the truth-table axioms in polynomial time. However, since exponentially hard f_n exist unconditionally, we are also able to get an unconditional equivalence for a non-uniform propositional proof system for which the truth tables of hard f_n are hardcoded into the verification procedure.

$\{F_k\}$ is a family containing a sequence of formulas encoding the correctness of circuits C for Discrete Logarithm, for each possible sequence of kn^k-size circuits C. The interesting direction of the equivalences in Theorem 1 follows from self-provability, while the other direction follows immediately from the soundness of the proof system Q.

We remark that some formulas in sequences F_k are not tautologies, indeed if Discrete Logarithm was hard, all sequences F_k would include non-tautologies. Similar recent results about strong proof systems [48, 53] also involve formulas that are possibly non-tautologies. We remark that this seems unavoidable in the context of strong proof systems, since formulas that we *know* to be tautologies also tend to have *short proofs* in strong proof systems, so we are unlikely to be able to say anything interesting about their hardness. Moreover, if we try to prove a proof complexity lower bound in order to derive a strong circuit lower bound using the connection in Theorem 1, we can assume for free that the lower bound we are proving is for a tautology, as otherwise the lower bound holds trivially by soundness of the proof system.

Witnessing $\mathsf{NP} \not\subseteq \mathsf{P/poly}$. We next consider whether circuit upper bounds for SAT are self-provable. This corresponds to feasible witnessing of $\mathsf{SAT} \notin \mathsf{P/poly}$. The advantage of self-provability for SAT is that it would allow us to argue that strong circuit lower bounds follow from proof complexity lower bounds for *any* sequence of tautologies for some concrete proof system Q. In contrast, the connection in Theorem 1 is for a specific family of formulas.

Similar kinds of witnessing for SAT have been considered before in the literature, using diagonalization techniques [25, 5, 10]. Indeed, Bogdanov, Talwar and Wan [10] call a similar feasible witnessing in the uniform setting a "dreambreaker" (citing Adam Smith) and show that such a feasible witnessing can be constructed. However, in their result, there is no guarantee on the lengths of counter-examples that are produced by the witnessing function. In our applications, it is crucial that the witnessing function finds an error on the input length of the given circuit assuming just that the circuit errs on the given input length.

A witnessing function f for SAT (described at the beginning of Section 1.2) exists under the assumption of the existence of a one-way function and a function in E hard for subexponential-size circuits [46, 44], as formalized in Lemma 2 in this paper. Is it however possible to construct it without assuming more than the assumption we want to witness? Formally, we are asking if there is a p-time function f such that for each big enough n propositional formula $w_n^k(f)$, defined by

$$w_n^k(f) := [\mathsf{SAT}_n(x,y) \to \mathsf{SAT}_n(x, C(x))] \vee [\mathsf{SAT}_n(f_1(C), f_2(C)) \wedge \neg \mathsf{SAT}_n(f_1(C), C(f_1(C)))],$$

is a tautology. Here, $\mathsf{SAT}_n(x,y)$ is a p-time predicate saying that x is an n-bit string encoding a propositional formula satisfied by assignment y, $C(z)$ says that free variables C represent a circuit with n inputs, $\leq n$ outputs and size n^k which outputs $C(z)$ on z, and $f(C)$ outputs a pair of strings $\langle f_1(C), f_2(C)\rangle$.

Intuitively, $w_n^k(f)$ states that for all formulas x of length n and circuits C of size n^k on n inputs, either C finds a satisfying assignment to every satisfiable x, or the witnessing function f outputs a "counter-example" $f_1(C)$ together with a satisfying assignment $f_2(C)$ to $f_1(C)$ such that C does not output a correct witness to $f_1(C)$. We do not specify the precise encoding of the formula $w_n^k(f)$. The proof systems we work with simulate EF and are therefore strong enough to reason efficiently with any natural encoding of $w_n^k(f)$.

If we had a function f such that for some n_0 and all $n > n_0$, $w_n^k(f)$ would be a tautology, we could define an extension of EF, denoted $\mathsf{EF} + w^k(f)$, such that $\mathsf{EF} + w^k(f)$ proofs are EF-proofs which are, in addition, allowed to derive substitutional instances of $w_n^k(f)$, for $n > n_0$.

We prove the following theorem. Formulas $W_{n_0}^k(f)$ in the statement of Theorem 2 denote a natural $\forall\Pi_1^b$-formalization of the statement "$\forall n > n_0, w_n^k(f)$", see §2.2 for the definition of Π_1^b. By the correspondence between S_2^1 and EF, cf. [38], if $W_{n_0}^k(f)$ was provable in S_2^1, for some p-time f, then tautologies $w_n^k(f)$, for $n \geq n_0$, would have p-size proofs in EF.

Theorem 2 (Circuit complexity from proof complexity & witnessing of $\mathsf{NP} \not\subseteq \mathsf{P/poly}$)**.** *Let $k \geq 1$ be a constant.*

1. *Suppose that there is a p-time function f such that for each big enough n, $w_n^k(f)$ is a tautology. If $\mathsf{EF} + w^k(f)$ is not p-bounded, then $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^{\epsilon k}]$ for infinitely many n.*

2. *Suppose that there is a p-time function f such that for some n_0, $\mathsf{S}_2^1 \vdash W_{n_0}^k(f)$. If EF is not p-bounded, then $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^{\epsilon k}]$ for infinitely many n.*

In Items 1 and 2, $\epsilon > 0$ is a universal constant (independent of k).

The idea of the proof of Theorem 2 is to use the witnessing tautologies to argue self-provability for SAT. Item 2 uses the standard result that EF is the propositional translation of Buss's theory S_2^1.

Notably, if for all $k \geq 1$ there is a p-time function f^k such that for each big enough n, $w_n^k(f^k)$ is a tautology, then $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$. This follows from Theorem 2 and Proposition 1.

Corollary 1 (Circuit lower bounds from witnessing).
If for all $k \geq 1$ there is a p-time function f^k such that for each big enough n, $w_n^k(f^k)$ is a tautology, then $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

Nonuniform witnessing. If we allow f to be nonuniform, we obtain a version of formulas $w_n^k(f)$ which are unconditionally tautological. This follows from a theorem of Lipton and Young [42], who showed that for each sufficiently big n and each Boolean function f with n inputs which is hard for circuits of size s^3, $s \geq n^3$, there is a set $A_n^{f,s} \subseteq \{0,1\}^n$ of size $poly(s)$ such that no s-size circuit computes f on $A_n^{f,s}$. The set $A_n^{f,s}$ is the set of *anticheckers* of f w.r.t. s. Let n be sufficiently big and α_n^s be tautologies defined by

$$\alpha_n^s := \big(\mathsf{SAT}_n(x,y) \rightarrow \mathsf{SAT}_n(x, B(x))\big) \vee \Big(\bigvee_{z \in A} C(z) \neq \mathsf{SAT}_n(z)\Big),$$

where $\mathsf{SAT}_n(z) \in \{0,1\}$ is such that $\mathsf{SAT}_n(z) = 1 \Leftrightarrow \exists y, \mathsf{SAT}_n(z,y)$. A is $A_n^{\mathsf{SAT}_n,s}$ if $\mathsf{SAT}_n \notin \mathsf{Circuit}[s^3]$ and an arbitrary $poly(s)$-size subset of $\{0,1\}^n$ otherwise. $C(z)$ in the right disjunct of α_n^s stands for the output of a single-output s-size circuit C with input z. The circuit C is represented by free variables. The circuit B in the left disjunct is a fixed $poly(s)$-size circuit, with n inputs and $\leq n$ outputs, obtained from a fixed s^3-size single-output circuit B' with n inputs such that

$$\mathsf{SAT}_n \in \mathsf{Circuit}[s^3] \Leftrightarrow \forall x \in \{0,1\}^n, B'(x) = \mathsf{SAT}_n(x).$$

The circuit B' exists by considering two cases: I. If $\mathsf{SAT}_n \notin \mathsf{Circuit}[s^3]$, we can take arbitrary s^3-size circuit B'; II. If $\mathsf{SAT}_n \in \mathsf{Circuit}[s^3]$, we can let B' be an s^3-size circuit computing SAT_n. B is obtained from B' in a standard way so that B solves the search version of SAT_n if $\mathsf{SAT}_n \in \mathsf{Circuit}[s^3]$. Analogously to Theorem 2, we get the following.

Theorem 3 (Circuit complexity from nonuniform proof complexity).
Let $k \geq 3$ be a constant. If there are tautologies without p-size EF*-derivations from substitutional instances of tautologies $\alpha_n^{n^k}$, then* $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^k]$ *for infinitely many n.*

Theorem 3 shows that $\mathsf{EF} + \alpha^{n^k}$, defined analogously as $\mathsf{EF} + w^k(f)$, is in certain sense optimal: If $\forall n, \mathsf{SAT}_n \in \mathsf{Circuit}[n^k]$, then $\mathsf{EF} + \alpha^{n^k}$ has $poly(n)$-size proofs of all tautologies. In fact, there is a p-size circuit which given a tautology ϕ of size n outputs its proof in $\mathsf{EF}+\alpha^{n^k}$. Cook and Krajíček [21] constructed an optimal proof system with 1 bit of nonuniform advice. Their system differs from $\mathsf{EF} + \alpha^{n^k}$ in that it is based on a diagonalization simulating all possible proof systems.

1.3 Metamathematical Implications

We use the results from the previous section to shed light on some classical questions in complexity theory. Specifically, we show that provability of efficient anticheckers for SAT, of efficient reductions collapsing Impagliazzo's worlds Heuristica and Pessiland, or of efficient reductions from the non-existence of one-way functions to learning, would imply new connections between proof complexity and circuit complexity.

I. Feasible anticheckers.

Below, we use 'CC ← PC' to abbreviate that strong circuit lower bounds follow from proof complexity lower bounds.

Theorem 4 ('CC ← PC' from feasible anticheckers - Informal, cf. Theorem 10)**.** *Let $k \geq 3$ be a constant and assume that there is a p-time function f such that S^1_2 proves that for all integers n either $f(1^n)$ outputs a $poly(n)$ circuit B solving* SAT *on inputs of length n, or $f(1^n)$ outputs an antichecker of* SAT *on inputs of length n with respect to size n^k, together with satisfying assignments for all YES instances of the antichecker.*

Then, EF *is not p-bounded implies* $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^k]$ *for infinitely many n.*

Theorem 4 follows from the proof of Theorem 2, see §4.

Similarly as in Corollary 1, if for each $k \geq 3$ there is a p-time function f such that S^1_2 proves the statement about the existence of anticheckers from the assumption of Theorem 4, then $\mathsf{NEXP} \not\subseteq \mathsf{P/poly}$.

II. One-way functions from $\mathsf{NP} \not\subseteq \mathsf{P/poly}$.

Denote by $\mathsf{tt}(f_n, s)$ a propositional formula expressing that Boolean function $f_n : \{0,1\}^n \mapsto \{0,1\}$ represented by its truth-table is not computable by a Boolean circuit of size s represented by free variables, see §2.4. So $\mathsf{tt}(f_n, s)$ is a tautology if and only if f_n is hard for circuits of size s. The size of the formula $\mathsf{tt}(f_n, s)$ is $poly(2^n, s)$. Similarly, let $\mathsf{tt}(f_n, s, t)$ be a formula expressing that circuits of size s fail to compute f on $\geq t$-fraction of inputs.

Given a function $h \in \mathsf{E}$ such that for some n_0, for each $n \geq n_0$, each $s(2^n)$-size circuit with n inputs fails to compute h_n on $\geq t(2^n)$-fraction of inputs, where s, t are p-time functions in 2^n, we define a proof system $\mathsf{EF} + \mathsf{tt}(h, s, t)$ as an extension of EF which is allowed to derive in its proofs substitutional instances of $\mathsf{tt}(h_n, s(2^n), t(2^n))$, for $n \geq n_0$.

Theorem 5 ('CC ← PC' from 'OWF ← $\mathsf{NP} \not\subseteq \mathsf{P/poly}$' & hard E - Informal, cf. Theorem 11)**.** *Assume that there is $h' \in \mathsf{E}$ such that for each sufficiently big n, each $2^{n/4}$-size circuit fails to compute h' on $\geq 1/2 - 1/2^{n/4}$ fraction of all inputs of length n. Further assume that there are p-time functions h and f such that S^1_2 proves that for each integer n and each fixed-polynomial size circuit C, $f(C)$ either outputs a $poly(n)$-size circuit B solving* SAT*, or C fails to invert h with significant probability over a uniformly random input to h. Then, if* $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ *is not p-bounded, we have that* $\mathsf{SAT} \notin \mathsf{P/poly}$.

We remark that if there is a p-time function f' and constant n_0 such that S^1_2 proves that for each $n \geq n_0$, $f'(1^{2^n})$ outputs the truth-table of a function h' with n inputs which is hard on average for $2^{n/4}$-size circuits, then EF and $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ are p-equivalent. (Note that $h' \in \mathsf{E}$.)

The proof of Theorem 5 is based on a formalization of the already-mentioned fact that given a one-way function h and a function in E hard for subexponential-size circuits, we can construct a p-time function witnessing errors of p-size circuits attempting to solve SAT. (The witnessing function outputs formulas encoding the statement $h(x) = b$, with free variables x and suitable constants b, cf. Lemma 2.) We formalize the conditional witnessing in a theory $\mathsf{S}^1_2 + dWPHP(\mathsf{PV})$, where $dWPHP(\mathsf{PV})$ stands for a dual weak pigeonhole principle, see §2.2. Combining this with the assumption that S^1_2 proves that a one-way function can be obtained from the hardness of SAT, we obtain the 'ideal' $(\mathsf{S}^1_2 + dWPHP(\mathsf{PV}))$-provable witnessing similar to the tautology $w^k_n(f)$. Having the ideal witnessing statement we proceed as in the proof of Theorem 2 with the difference that the axiom $dWPHP(\mathsf{PV})$ and the assumed hardness of E lead to the system $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ instead of EF, see §5.

III. Learning from the non-existence of OWFs.

Theorem 6 ('CC ← PC' from 'Learning ← $\nexists$ OWF' & hardness of E - Informal, cf. Theorem 12)**.** *Let $k \geq 1$ be an arbitrary constant. Assume that there is $h' \in \mathsf{E}$ such that for each sufficiently big n, each $2^{n/4}$-size circuit fails to compute h' on $\geq 1/2 - 1/2^{n/4}$ fraction of all inputs of length n. Further assume that there are p-time functions h and f such that S^1_2 proves that for all integers n and fixed-polynomial size circuits C, either $f(C)$ outputs an efficient circuit B that weakly learns Boolean circuits with membership queries over the uniform distribution, or C fails to invert h with significant probability.*

Then the existence of a function $g_n : \{0,1\}^n \mapsto \{0,1\}$ that is average-case hard for polynomial-size circuits and such that $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ does not have a poly-size proof of the hardness of g_n (represented by truth table formulas) implies that $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^k]$.

Theorem 6 is proved similarly as Theorem 5 with the difference that the provability of efficient learning allows us to prove efficiently only circuit lower bounds instead of all tautologies, see §6.

[47, Lemma 4] shows that, assuming E is S^1_2-provably hard as in Theorem 6, learning algorithms for small circuits can be S^1_2-provably constructed from circuits automating EF on tt-formulas.[4] Theorem 6 thus implies that (assuming E is S^1_2-provably hard) S^1_2-deriving automatability of EF from the non-existence of one-way functions would reduce circuit complexity to EF lower bounds.

[4] [47, Lemma 4] assumes also the existence of a prime. The assumption can be removed after moving to the propositional setting.

Generalization to stronger proof systems. S^1_2 in Theorems 2 & 4-6 can be replaced by essentially an arbitrary first-order theory T containing S^1_2 and satisfying some basic properties, if we simultaneously replace EF in conclusions of Theorems 2 & 4-6 by a suitable propositional proof system P_T such that propositional translations of Π^b_1 theorems of T have p-size proofs in P_T.

Weakening the assumptions. The core component of Theorems 2 & 4-6 is the existence of a suitable reduction. For example, in case of Theorem 6 we need a p-time reduction constructing learning algorithms from circuits breaking one-way functions. If such a reduction exists, even without assuming its provability in S^1_2, we can build a propositional proof system P by adding tautologies encoding the correctness of the reduction to EF. Then, showing that the resulting proof system $P + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ is not p-bounded on tt-tautologies would separate P and NP. This shows that the S^1_2-provability in the assumptions of our theorems can be weakened just to the validity of the respective statements, if we use stronger systems than EF in their conclusions. It also shows that most of the technicalities in the present paper stem from making the presented approach work for EF.

This has interesting metamathematical implications, in that provability of several important open questions in complexity theory within *any* system of bounded arithmetic would yield new connections between proof complexity and circuit complexity.

Moreover, we remark that if our final goal is to prove that $\mathsf{P} \neq \mathsf{NP}$, then the first assumption of Theorems 5-6 postulating the existence of a hard Boolean function in E is given to us 'for free', as otherwise, if all functions in E can be approximated by subexponential-size circuits, it is not hard to show that $\mathsf{P} \neq \mathsf{NP}$.

1.3.1 Plausibility of the assumptions

Impagliazzo's worlds. In a famous survey of Impagliazzo [28], he described 5 possible worlds of average-case complexity: Algorithmica, Heuristica, Pessiland, Minicrypt and Cryptomania. Recently, there have been various approaches proposed to rule out Heuristica and Pessiland (see, for example, [27, 52, 43]) by studying the complexity of problems about compression, such as the Minimum Circuit Size Problem (MCSP) and the problem of computing time-bounded Kolmogorov complexity. Our results have implications for the feasibility of such efforts - provable collapses of Impagliazzo's worlds would imply a new and surprising link between proof complexity and circuit complexity. For example, the reduction assumed in Theorem 6 asks for a construction of learning algorithms from circuits breaking one-way functions. Morally, the existence of such a reduction would rule out Pessiland out of Impagliazzo's worlds. Similarly, the reduction assumed in Theorem 5 would rule out Pessiland and Heuristica. The question of basing one-way functions on the worst-case hardness of NP has received much attention, and there are barriers known for restricted black-box reductions [16, 2, 15]. However, the reductions we consider are *white-box*, and we only require provability of the reduction.

Feasible MinMax theorem. The proofs of the existence of antichekers we are aware of use the efficient MinMax theorem [3, 42, 45] or similar methods. If we had a proof of MinMax which would use counting with only polynomial precision (formally, $\mathsf{APC_1}$-counting) and if we could replace p-time f in Theorem 4 by the existential quantifiers (see §4 for a discussion of the issue), we could prove the existence of anticheckers in $\mathsf{APC_1}$ and obtain the desired reduction of circuit complexity to proof complexity. Here, $\mathsf{APC_1}$ is Jeřábek's theory of approximate counting, see §2.2.

S^1_2-provability of a circuit lower bound. If we want to replace $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ in Theorems 5 & 6 by EF, it suffices to assume the S^1_2-provability of a subexponential circuit lower bound for E. This assumption has an interesting status. Razborov's conjecture about hardness of Nisan-Wigderson generators implies a conditional hardness of formulas $\mathsf{tt}(h, n^{O(1)})$ for Frege (for every h), cf. [51], and it is possible to consider extensions of the conjecture to all standard proof systems, even set theory ZFC. A conditional hardness of tt-formulas (for EF) follows also from a conjecture of Krajíček [36, Conjecture 7.9]. If the tt-formulas expressing subexponential lower bounds for E are hard for EF, then S^1_2 cannot prove the lower bounds either. On the other hand, it is not known how to prove hardness of $\mathsf{tt}(h, 2^{n/4})$, for all h, for Frege, under any standard complexity-theoretic hardness assumption. Moreover, all major circuit lower bounds for weak circuit classes and explicit Boolean functions are known to be provable in S^1_2, cf. [50, 44].[5] It is thus perfectly possible that subexponential average-case circuit lower bounds for E are provable in a theory such as S^1_2.[6]

1.3.2 Revising the status of the Cook-Reckhow program

Showing that statements like $\mathsf{P} \neq \mathsf{NP}$ follow from proof complexity lower bounds for concrete proof systems is considered so challenging that there have not been practically any conscious attempts to approach it. Our results show that the significant efforts that have been made in order to address some of the central problems in cryptography and learning theory are, in fact, aiming to establish precisely that. This can be interpreted as an evidence for the hardness of resolving the relevant problems in cryptography and learning theory, but alternatively as a step toward realizing the Cook-Reckhow program successfully. In any case, the presented results demonstrate a new fundamental connection between proof complexity, cryptography and learning theory.

[5] This has not been verified for lower bounds obtained via the algorithmic method of Williams [54].

[6] We emphasize that Theorems 5 & 6 do not require that S^1_2 proves a circuit lower bound. Further, we can replace the system $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ in Theorems 5 & 6 by, say, a propositional proof system corresponding to ZFC, if we assume that a subexponential circuit lower bound for E is provable in (the set theory) ZFC.

2 Preliminaries

2.1 Learning algorithms

$[n]$ denotes $\{1, \ldots, n\}$. 1^n stands for a string of n 1s. $\mathsf{Circuit}[s]$ denotes fan-in two Boolean circuits of size at most s. The size of a circuit is the number of its gates. A function $f : \{0,1\}^n \mapsto \{0,1\}$ is γ-approximated by a circuit C, if $\Pr_x[C(x) = f(x)] \geq \gamma$.

Definition 1 (PAC learning)**.** *A circuit class $\mathcal{C}$ is learnable over the uniform distribution by a circuit class $\mathcal{D}$ up to error ϵ with confidence δ, if there are randomized oracle circuits L^f from $\mathcal{D}$ such that for every Boolean function $f : \{0,1\}^n \mapsto \{0,1\}$ computable by a circuit from $\mathcal{C}$, when given oracle access to f, input 1^n and the internal randomness $w \in \{0,1\}^*$, L^f outputs the description of a circuit satisfying*

$$\Pr_w[L^f(1^n, w)\ (1-\epsilon)\text{-}\mathit{approximates}\ f] \geq \delta.$$

L^f uses non-adaptive membership queries if the set of queries which L^f makes to the oracle does not depend on the answers to previous queries. L^f uses random examples if the set of queries which L^f makes to the oracle is chosen uniformly at random.

In this paper, PAC learning always refers to learning over the uniform distribution. While, a priori, learning over the uniform distribution might not reflect real-world scenarios very well (and on the opposite end, learning over all distributions is perhaps overly restrictive), as far as we can tell it is possible that PAC learning of p-size circuits over the uniform distribution implies PAC learning of p-size circuits over all p-samplable distributions. Binnendyk, Carmosino, Kolokolova, Ramyaa and Sabin [9] proved the implication, if the learning algorithm in the conclusion is allowed to depend on the p-samplable distribution.

2.2 Bounded arithmetic and propositional logic

Theories of bounded arithmetic capture various levels of feasible reasoning and present a uniform counterpart to propositional proof systems.

The first theory formalizing p-time reasoning was introduced by Cook [19] as an equational theory PV. We work with its first-order conservative extension PV_1 from [39]. The language of PV_1, denoted PV as well, consists of symbols for all p-time algorithms given by Cobham's characterization of p-time functions, cf. [18]. A PV-formula is a first-order formula in the language PV. Σ^b_0 ($=\Pi^b_0$) denotes PV-formulas with only sharply bounded quantifiers $\exists x, x \leq |t|$, $\forall x, x \leq |t|$, where $|t|$ is "the length of the binary representation of t". Inductively, Σ^b_{i+1} resp. Π^b_{i+1} is the closure of Π^b_i resp. Σ^b_i under positive Boolean combinations, sharply bounded quantifiers, and bounded quantifiers $\exists x, x \leq t$ resp. $\forall x, x \leq t$.

Predicates definable by Σ_i^b resp. Π_i^b formulas are in the Σ_i^p resp. Π_i^p level of the polynomial hierarchy, and vice versa. PV_1 is known to prove $\Sigma_0^b(\mathsf{PV})$-induction:

$$A(0) \wedge \forall x\ (A(x) \rightarrow A(x+1)) \rightarrow \forall x A(x),$$

for Σ_0^b-formulas A, cf. Krajíček [34].

Buss [13] introduced the theory S_2^1 extending PV_1 with the Σ_1^b-length induction:

$$A(0) \wedge \forall x < |a|, (A(x) \rightarrow A(x+1)) \rightarrow A(|a|),$$

for $A \in \Sigma_1^b$. S_2^1 proves the sharply bounded collection scheme $BB(\Sigma_1^b)$:

$$\forall i < |a|\ \exists x < a, A(i,x) \rightarrow \exists w\ \forall i < |a|, A(i,[w]_i),$$

for $A \in \Sigma_1^b$ ($[w]_i$ is the ith element of the sequence coded by w), which is unprovable in PV_1 under a cryptographic assumption, cf. [20]. On the other hand, S_2^1 is $\forall\Sigma_1^b$-conservative over PV_1. This is a consequence of Buss's witnessing theorem stating that $\mathsf{S}_2^1 \vdash \exists y, A(x,y)$ for $A \in \Sigma_1^b$ implies $\mathsf{PV}_1 \vdash A(x, f(x))$ for some PV-function f.

Following a work by Krajíček [36], Jeřábek [30, 31, 32] systematically developed a theory APC_1 capturing probabilistic p-time reasoning by means of approximate counting.[7] The theory APC_1 is defined as $\mathsf{PV}_1 + dWPHP(\mathsf{PV})$ where $dWPHP(\mathsf{PV})$ stands for the dual (surjective) pigeonhole principle for PV-functions, i.e. for the set of all formulas

$$x > 0 \rightarrow \exists v < x(|y|+1) \forall u < x|y|,\ f(u) \neq v,$$

where f is a PV-function which might involve other parameters not explicitly shown. We devote §2.3 to a more detailed description of the machinery of approximate counting in APC_1.

Any Π_1^b-formula ϕ provable in S_2^1 can be expressed as a sequence of tautologies $||\phi||_n$ with proofs in the Extended Frege system EF which are constructible in p-time (given a string of the length n), cf. [19]. We refer to Krajíček [38] for basic notions in proof complexity such as EF. As it is often easier to present a proof in a theory of bounded arithmetic than in the corresponding propositional system, bounded arithmetic functions, so to speak, as a uniform language for propositional logic.

2.3 Approximate counting

In order to prove our results we will need to use Jeřábek's theory of approximate counting. This section recalls the properties of APC_1 we will need.

[7] Krajíček [36] introduced a theory BT defined as $\mathsf{S}_2^1 + dWPHP(\mathsf{PV})$ and proposed it as a theory for probabilistic p-time reasoning.

By a definable set we mean a collection of numbers satisfying some formula, possibly with parameters. When a number a is used in a context which asks for a set it is assumed to represent the integer interval $[0,a)$, e.g. $X\subseteq a$ means that all elements of set X are less than a. If $X\subseteq a$, $Y\subseteq b$, then $X\times Y := \{bx+y \mid x\in X, y\in Y\}\subseteq ab$ and $X\dot{\cup}Y := X\cup\{y+a \mid y\in Y\}\subseteq a+b$. Rational numbers are assumed to be represented by pairs of integers in the natural way. We use the notation $x\in Log\leftrightarrow \exists y,\ x=|y|$ and $x\in LogLog \leftrightarrow \exists y,\ x=||y||$.

Let $C: 2^n\rightarrow 2^m$ be a circuit and $X\subseteq 2^n, Y\subseteq 2^m$ definable sets. We write $C: X\twoheadrightarrow Y$ if $\forall y\in Y \exists x\in X,\ C(x)=y$. Jeřábek [32] gives the following definitions in $\mathsf{APC_1}$, but they can be formulated in $\mathsf{PV_1}$ as well.

Definition 2. *Let $X,Y\subseteq 2^n$ be definable sets, and $\epsilon\leq 1$. The size of X is approximately less than the size of Y with error ϵ, written as $X\preceq_\epsilon Y$, if there exists a circuit C, and $v\neq 0$ such that*

$$C: v\times(Y\dot{\cup}\epsilon 2^n)\twoheadrightarrow v\times X.$$

$X\approx_\epsilon Y$ stands for $X\preceq_\epsilon Y$ and $Y\preceq_\epsilon X$.

Since a number s is identified with the interval $[0,s)$, $X\preceq_\epsilon s$ means that the size of X is at most s with error ϵ.

The definition of $X\preceq_\epsilon Y$ is an unbounded $\exists\Pi^b_2$-formula even if X,Y are defined by circuits so it cannot be used freely in bounded induction. Jeřábek [32] solved this problem by working in HARD^A, a conservative extension of $\mathsf{APC_1}$, defined as a relativized theory $\mathsf{PV_1}(\alpha)+dWPHP(\mathsf{PV}(\alpha))$ extended with axioms postulating that $\alpha(x)$ is a truth-table of a function on $||x||$ variables hard on average for circuits of size $2^{||x||/4}$, see §2.4.2. In HARD^A there is a $\mathsf{PV_1}(\alpha)$ function $Size$ approximating the size of any set $X\subseteq 2^n$ defined by a circuit C so that $X\approx_\epsilon Size(C,2^n,2^{\epsilon^{-1}})$ for $\epsilon^{-1}\in Log$, cf. [32, Lemma 2.14]. If $X\cap t\subseteq 2^{|t|}$ is defined by a circuit C and $\epsilon^{-1}\in Log$, we can define

$$\Pr_{x<t}[x\in X]_\epsilon := \frac{1}{t}Size(C,2^{|t|},2^{\epsilon^{-1}}).$$

The presented definitions of approximate counting are well-behaved:

Proposition 2 (Jeřábek [32]). *(in $\mathsf{PV_1}$) Let $X,X',Y,Y',Z\subseteq 2^n$ and $W,W'\subseteq 2^m$ be definable sets, and $\epsilon,\delta<1$. Then*

i) $X\subseteq Y\Rightarrow X\preceq_0 Y$,

ii) $X\preceq_\epsilon Y\wedge Y\preceq_\delta Z\Rightarrow X\preceq_{\epsilon+\delta} Z$,

iii) $X\preceq_\epsilon X'\wedge W\preceq_\delta W'\Rightarrow X\times W\preceq_{\epsilon+\delta+\epsilon\delta} X'\times W'$.

iv) $X\preceq_\epsilon X'\wedge Y\preceq_\delta Y'$ and X',Y' are separable by a circuit, then $X\cup Y\preceq_{\epsilon+\delta} X'\cup Y'$.

Proposition 3 (Jeřábek [32]). *(in $\mathsf{APC_1}$)*

1. Let $X,Y\subseteq 2^n$ be definable by circuits, $s,t,u\leq 2^n$, $\epsilon,\delta,\theta,\gamma<1,\gamma^{-1}\in Log$. Then

i) $X \preceq_\gamma Y$ *or* $Y \preceq_\gamma X$*,*
ii) $s \preceq_\epsilon X \preceq_\delta t \Rightarrow s < t + (\epsilon + \delta + \gamma)2^n$*,*
iii) $X \preceq_\epsilon Y \Rightarrow 2^n \backslash Y \preceq_{\epsilon+\gamma} 2^n \backslash X$*,*
iv) $X \approx_\epsilon s \wedge Y \approx_\delta t \wedge X \cap Y \approx_\theta u \Rightarrow X \cup Y \approx_{\epsilon+\delta+\theta+\gamma} s + t - u$*.*

2. (Disjoint union) Let $X_i \subseteq 2^n$*,* $i < m$ *be defined by a sequence of circuits and* $\epsilon, \delta \leq 1$*,* $\delta^{-1} \in Log$*. If* $X_i \preceq_\epsilon s_i$ *for every* $i < m$*, then* $\bigcup_{i<m}(X_i \times \{i\}) \preceq_{\epsilon+\delta} \sum_{i<m} s_i$*.*

When proving Σ^b_1 statements in $\mathsf{APC_1}$ we can afford to work in $\mathsf{S}^1_2 + dWPHP(\mathsf{PV}) + BB(\Sigma^b_2)$ and, in fact, assuming the existence of a single hard function in $\mathsf{PV_1}$ gives us the full power of $\mathsf{APC_1}$. Here, $BB(\Sigma^b_2)$ is defined as $BB(\Sigma^b_1)$ but with $A \in \Sigma^b_2$.

Lemma 1 ([44])**.** *Suppose* $\mathsf{S}^1_2 + dWPHP(\mathsf{PV}) + BB(\Sigma^b_2) \vdash \exists y A(x, y)$ *for* $A \in \Sigma^b_1$*. Then, for every* $\epsilon < 1$*, there is* k *and* PV*-functions* g, h *such that* $\mathsf{PV_1}$ *proves*

$$|f| \geq |x|^k \wedge \exists y, |y| = ||f||, C_h(y) \neq f(y) \to A(x, g(x, f))$$

where $f(y)$ *is the* y*th bit of* f*,* $f(y) = 0$ *for* $y > |f|$*, and* C_h *is a circuit of size* $\leq 2^{\epsilon||f||}$ *generated by* h *on* f, x*. Moreover,* $\mathsf{APC_1} \vdash \exists y A(x, y)$*.*

2.4 Formalizing complexity-theoretic statements

2.4.1 Circuit lower bounds

An ‘almost everywhere’ circuit lower bound for circuits of size s and a function f says that for every sufficiently big n, for each circuit C with n inputs and size s, there exists an input y on which the circuit C fails to compute $f(y)$.

If $f : \{0,1\}^n \to \{0,1\}$ is an NP function and $s = n^k$ for a constant k, this can be written down as a $\forall\Sigma^b_2$ formula $\mathsf{LB}(f, n^k)$,

$$\forall n,\ n > n_0\ \forall \text{ circuit } C \text{ of size } \leq n^k\ \exists y,\ |y| = n,\ C(y) \neq f(y),$$

where n_0 is a constant and $C(y) \neq f(y)$ is a Σ^b_2 formula stating that a circuit C on input y outputs the opposite value of $f(y)$. The intended meaning of ‘$\exists y, |y| = n$’ is to say that y is a string from $\{0,1\}^n$. This is a slight abuse of notation as, formally, $|y| = n$ fixes the first bit of y to 1.

If we want to express $s(n)$-size lower bounds for $s(n)$ as big as $2^{O(n)}$, we add an extra assumption on n stating that $\exists x,\ n = ||x||$. That is, the resulting formula $\mathsf{LB_{tt}}(f, s(n))$ has form ‘$\forall x, n; n = ||x|| \to \dots$’. Treating x, n as free variables, $\mathsf{LB_{tt}}(f, s(n))$ is Π^b_1 if f is, for instance, SAT because $n = ||x||$ implies that the quantifiers bounded by $2^{O(n)}$ are sharply bounded. Moreover, allowing $f \in \mathsf{NE}$ lifts the complexity of $\mathsf{LB_{tt}}(f, s(n))$ just to $\forall\Sigma^b_1$. The function $s(n)$ in $\mathsf{LB_{tt}}(f, s(n))$ is assumed to be a PV-function with input x (satisfying $||x|| = n$).

In terms of the *Log*-notation, $\mathsf{LB}(f, n^k)$ implicitly assumes $n \in Log$ while $\mathsf{LB}_{\mathsf{tt}}(f, n^k)$ assumes $n \in LogLog$. By chosing the scale of n we are determining how big objects are going to be 'feasible' for theories reasoning about the statement. In the case $n \in LogLog$, the truth-table of f (and everything polynomial in it) is feasible. Assuming just $n \in Log$ means that only the objects of polynomial-size in the size of the circuit are feasible. Likewise, the theory reasoning about the circuit lower bound is less powerful when working with $\mathsf{LB}(f, n^k)$ than with $\mathsf{LB}_{\mathsf{tt}}(f, n^k)$. (The scaling in $\mathsf{LB}_{\mathsf{tt}}(f, s)$ corresponds to the choice of parameters in natural proofs and in the formalizations by Razborov [50].)

We can analogously define formulas $\mathsf{LB}_{\mathsf{tt}}(f, s(n), t(n))$ expressing an average-case lower bound for f, where f is a free variable (with $f(y)$ being the yth bit of f and $f(y) = 0$ for $y > |f|$). More precisely, $\mathsf{LB}_{\mathsf{tt}}(f, s(n), t(n))$ generalizes $\mathsf{LB}_{\mathsf{tt}}(f, s(n))$ by saying that each circuit of size $s(n)$ fails to compute f on at least $t(n)$ inputs, for PV-functions $s(n), t(n)$. Since $n \in LogLog$, $\mathsf{LB}_{\mathsf{tt}}(f, s(n), t(n))$ is Π^b_1.

Propositional version. An $s(n)$-size circuit lower bound for a function $f : \{0,1\}^n \to \{0,1\}$ can be expressed by a $poly(2^n, s)$-size propositional formula $\mathsf{tt}(f, s)$,

$$\bigvee_{y \in \{0,1\}^n} f(y) \neq C(y)$$

where the formula $f(y) \neq C(y)$ says that an $s(n)$-size circuit C represented by $poly(s)$ variables does not output $f(y)$ on input y. The values $f(y)$ are fixed bits. That is, the whole truth-table of f is hard-wired in $\mathsf{tt}(f, s)$.

The details of the encoding of the formula $\mathsf{tt}(f, s)$ are not important for us as long as the encoding is natural because systems like EF considered in this paper can reason efficiently about them. We will assume that $\mathsf{tt}(f, s)$ is the formula resulting from the translation of Π^b_1 formula $\mathsf{LB}_{\mathsf{tt}}(h, s)$, where $n_0 = 0$, n, x are substituted after the translation by fixed constants so that $x = 2^{2^n}$, and h is a free variable (with $h(y)$ being the yth bit of h and $h(y) = 0$ for $y > |h|$) which is substituted after the translation by constants defining f.

Analogously, we can express average-case lower bounds by propositional formulas $\mathsf{tt}(f, s(n), t(n))$ obtained by translating $\mathsf{LB}_{\mathsf{tt}}(h, s(n), t(n)2^n)$, with $n_0 = 0$, fixed $x = 2^{2^n}$ and h substituted after the translation by f.

2.4.2 Learning algorithms

A circuit class $\mathcal{C}$ is defined by a PV-formula if there is a PV-formula defining the predicate $C \in \mathcal{C}$. Definition 1 can be formulated in the language of HARD^A: A circuit class $\mathcal{C}$ (defined by a PV-formula) is learnable over the uniform disribution by a circuit class $\mathcal{D}$ (defined by a PV-formula) up to error ϵ with confidence δ, if there are randomized oracle circuits L^f from $\mathcal{D}$ such that for every Boolean function $f : \{0,1\}^n \mapsto \{0,1\}$ (represented by its truth-table) computable by a circuit from $\mathcal{C}$, for each $\gamma^{-1} \in Log$, when given oracle access to f, input 1^n and the internal randomness $w \in \{0,1\}^*$, L^f outputs the description of a circuit satisfying

$$\Pr_w[L^f(1^n,w)\ (1-\epsilon)\text{-approximates } f]_\gamma \geq \delta.$$

The inner probability of approximability of f by $L^f(1^n,w)$ is counted exactly. This is possible because f is represented by its truth-table, which implies that $2^n \in Log$.

Propositional version. In order to translate the definition of learning algorithms to propositional formulas and to the language of PV_1 we need to look more closely at the definition of HARD^A.

PV_1 can be relativized to $\mathsf{PV}_1(\alpha)$. The new function symbol α is then allowed in the inductive clauses for introduction of new function symbols. This means that the language of $\mathsf{PV}_1(\alpha)$, denoted also $\mathsf{PV}(\alpha)$, contains symbols for all p-time oracle algorithms.

Proposition 4 (Jeřábek [30]). *For every constant $\epsilon < 1/3$ there exists a constant n_0 such that APC_1 proves: for every $n \in LogLog$ such that $n > n_0$, there exist a function $f : 2^n \to 2$ such that no circuit of size $2^{\epsilon n}$ computes f on $\geq (1/2 + 1/2^{\epsilon n})2^n$ inputs.*

Definition 3 (Jeřábek [30]). *The theory HARD^A is an extension of the theory $\mathsf{PV}_1(\alpha) + dWPHP(\mathsf{PV}(\alpha))$ by the axioms*

1. *$\alpha(x)$ is a truth-table of a Boolean function in $||x||$ variables,*
2. *$\mathsf{LB}_{\mathsf{tt}}(\alpha(x), 2^{||x||/4}, 2^{||x||}(1/2 - 1/2^{||x||/4}))$, for constant n_0 from Proposition 4,*
3. *$||x|| = ||y|| \to \alpha(x) = \alpha(y)$.*

By inspecting the proof of Lemma 2.14 in [32], we can observe that on each input $C, 2^n, 2^{\epsilon^{-1}}$ the $\mathsf{PV}_1(\alpha)$-function $Size$ calls α just once (to get the truth-table of a hard function which is then used as the base function of the Nisan-Wgiderson generator). In fact, $Size$ calls α on input x which depends only on $|C|$, the number of inputs of C and w.l.o.g. also just on $|\epsilon^{-1}|$ (since decreasing ϵ leads only to a better approximation). In combination with the fact that the approximation $Size(C, 2^n, 2^{\epsilon^{-1}}) \approx_\epsilon X$, for $X \subseteq 2^n$ defined by C, is not affected by a particular choice of the hard boolean function generated by α, we get that APC_1 proves

$$\mathsf{LB}_{\mathsf{tt}}(y, 2^{||y||/4}, 2^{||y||}(1/2 - 1/2^{||y||/4})) \wedge ||y|| = S(C, 2^n, 2^{\epsilon^{-1}}) \to\ Sz(C, 2^n, 2^{\epsilon^{-1}}, y) \approx_\epsilon X,$$

where Sz is defined as $Size$ with the only difference that the call to $\alpha(x)$ on $C, 2^n, 2^{\epsilon^{-1}}$ is replaced by y and $S(C, 2^n, 2^{\epsilon^{-1}}) = ||x||$ for a PV-function S. (S is given by a subcomputation of $Size$ specifying $||x||$, for x on which $Size$ queries $\alpha(x)$.)

This allows us to express $\Pr_{x<t}[x \in X]_\epsilon = a$, where $\epsilon^{-1} \in Log$ and $X \cap t \subseteq 2^{|t|}$ is defined by a circuit C, without a $\mathsf{PV}_1(\alpha)$ function, by formula

$$\forall y\ (\mathsf{LB}_{\mathsf{tt}}(y, 2^{||y||/4}, 2^{||y||}(1/2 - 1/2^{||y||/4})) \wedge ||y|| = S(C, 2^{|t|}, 2^{\epsilon^{-1}}) \to Sz(C, 2^{|t|}, 2^{\epsilon^{-1}}, y)/t = a).$$

We denote the resulting formula by $\Pr^{y}_{x<t}[x \in X]_\epsilon = a$. We will use the notation $\Pr^{y}_{x<t}[x \in X]_\epsilon$ in equations with the intended meaning that the equation holds for the value $Sz(\cdot,\cdot,\cdot,\cdot)/t$ under corresponding assumptions. For example, $t \cdot \Pr^{y}_{x<t}[x \in X]_\epsilon \preceq_\delta a$ stands for '$\forall y, \exists v, \exists$ circuit $\hat{C}$ (defining a surjection) which witnesses that $\mathsf{LB}_{\mathsf{tt}}(y, 2^{||y||/4}, 2^{||y||}(1/2 - 1/2^{||y||/4})) \wedge ||y|| = S(C, 2^{|t|}, 2^{\epsilon^{-1}})$ implies $Sz(C, 2^{|t|}, 2^{\epsilon^{-1}}, y) \preceq_\delta a$'.

The definition of learning can be now expressed without a $\mathsf{PV}_1(\alpha)$ function: If circuit class $\mathcal{C}$ is defined by a PV-function, the statement that a given oracle algorithm L (given by a PV-function with oracle queries) learns a circuit class $\mathcal{C}$ over the uniform distribution up to error ϵ with confidence δ can be expressed as before with the only difference that we replace $\Pr_w[L^f(1^n, w)\ (1-\epsilon)\text{-approximates } f]_\gamma \geq \delta$ by

$$\Pr_{w}^{y}[L^f(1^n, w)\ (1-\epsilon)\text{-approximates } f]_\gamma \geq \delta.$$

Since the resulting formula A defining learning is not Π^b_1 (because of the assumption $\mathsf{LB}_{\mathsf{tt}}$) we cannot translate it to propositional logic. We will sidestep the issue by translating only the formula B obtained from A by deleting subformula $\mathsf{LB}_{\mathsf{tt}}$ (but leaving $||y|| = S(\cdot,\cdot,\cdot)$ intact) and replacing the variables y by fixed bits representing a hard boolean function. In more detail, Π^b_1 formula B can be translated into a sequence of propositional formulas $\mathsf{lear}^y_\gamma(L, \mathcal{C}, \epsilon, \delta)$ expressing that "if $C \in \mathcal{C}$ is a circuit computing f, then L querying f generates a circuit D such that $\Pr[D(x) = f(x)] \geq 1 - \epsilon$ with probability $\geq \delta$, which is counted approximately with precision γ". Note that C, f are represented by free variables and that there are also free variables for error γ from approximate counting and for Boolean functions y. As in the case of tt-formulas, we fix $|f| = 2^n$, so n is not a free variable. Importantly, $\mathsf{lear}^y_\gamma(L, \mathcal{C}, \epsilon, \delta)$ does not postulate that y is a truth-table of a hard boolean function. Nevertheless, for any fixed (possibly non-uniform) bits representing a sequence of Boolean functions $h = \{h_m\}_{m>n_0}$ such that h_m is not $(1/2 + 1/2^{m/4})$-approximable by any circuit of size $2^{m/4}$, we can obtain formulas $\mathsf{lear}^h_\gamma(L, \mathcal{C}, \epsilon, \delta)$ by substituting bits h for y.

Using a single function h in $\mathsf{lear}^h_\gamma(L, \mathcal{C}, \epsilon, \delta)$ does not ruin the fact that (the translation of function) Sz approximates the respective probability with accuracy γ because Sz queries a boolean function y which depends just on the number of atoms representing γ^{-1} and on the size of the circuit D defining the predicate we count together with the number of inputs of D. The size of D and the number of its inputs are w.l.o.g. determined by the number of inputs of f.

If we are working with formulas $\mathsf{lear}^h_\gamma(L, \mathcal{C}, \epsilon, \delta)$, where h is a sequence of bits representing a hard boolean function, in a proof system which cannot prove efficiently that h is hard, our proof system might not be able to show that the definition is well-behaved - it might not be able to derive some standard properties of the function Sz used inside the formula. Nevertheless, in our theorems this will never be the case: our proof systems will always know that h is hard.

In formulas $\mathsf{lear}^y_\gamma(L,\mathcal{C},\epsilon,\delta)$ we can allow L to be a sequence of nonuniform circuits, with a different advice string for each input length. One way to see that is to use additional input to L in Π^b_1 formula B, then translate the formula to propositional logic and substitute the right bits of advice for the additional input. Again, the precise encoding of the formula $\mathsf{lear}^y_\gamma(L,\mathcal{C},\epsilon,\delta)$ does not matter very much to us but in order to simplify proofs we will assume that $\mathsf{lear}^y_\gamma(L,\mathsf{Circuit}[n^k],\epsilon,\delta)$ has the from $\neg\mathsf{tt}(f,n^k)\rightarrow R$, where n,k are fixed, f is represented by free variables and R is the remaining part of the formula expressing that L generates a suitable circuit with high probability.

2.4.3 Nisan-Wigderson generators

The core theorem underlying approximate counting in APC_1 is the following formalization of Nisan-Wigderson generators (NW), cf. [30, Proposition 4.7].

Theorem 7 (Jeřábek [30])**.** *Let $0<\gamma<1$. There are constants $c>1$ and $\delta'>0$ such that for each $\delta<\delta'$ there is a poly(2^m)-time function*

$$NW:\{0,1\}^{cm}\times\{0,1\}^{2^m}\mapsto\{0,1\}^{\lfloor 2^{\delta m}\rfloor}$$

such that S^1_2 proves: "If $2^m\in Log$ and $f:\{0,1\}^m\mapsto\{0,1\}$ is a Boolean function such that no circuit of size $2^{\epsilon m}$ computes f on $>(1/2+1/2^{\epsilon m})2^m$ inputs, then for each $(2^{\epsilon m}-\lceil 2^{(\delta+\gamma)m}\rceil-1)$-size circuit D with $\lfloor 2^{\delta m}\rfloor$ inputs,

$$2^{\lfloor 2^{\delta m}\rfloor}\times\{z<2^{cm}\mid D(NW_f(z))=1\}\succeq_e 2^{cm}\times\{x<2^{\lfloor 2^{\delta m}\rfloor}\mid D(x)=1\},$$

where $e:=\lceil 2^{\delta m}\rceil/2^{\epsilon m}$ and $NW_f(z):=NW(z,f)$."

Theorem 7 shows that $\Pr^y_x[D(x)=1]_\theta$ is S^1_2-provably similar to $\Pr^y_z[D(NW_f(z))=1]_\theta$, for $\theta^{-1}\in Log$. To see this, note that

$$2^{cm}\Pr^y_z[D(NW_f(z))=1]_\theta\approx_\theta\{z<2^{cm}\mid D(NW_f(z))=1\}.$$

Hence, by Proposition 2 *iii*),

$$2^{\lfloor 2^{\delta m}\rfloor}2^{cm}\Pr^y_z[D(NW_f(z))=1]_\theta\succeq_\theta 2^{\lfloor 2^{\delta m}\rfloor}\times\{z<2^{cm}\mid D(NW_f(z))=1\}.$$

Similarly,

$$2^{cm}\times\{x<2^{\lfloor 2^{\delta m}\rfloor}\mid D(x)=1\}\succeq_\theta 2^{\lfloor 2^{\delta m}\rfloor}2^{cm}\Pr^y_x[D(x)=1]_\theta.$$

Therefore, by Proposition 2 *ii*), the conclusion of Theorem 7 implies

$$2^{\lfloor 2^{\delta m}\rfloor}2^{cm}\Pr^y_z[D(NW_f(z))=1]_\theta\succeq_{2\theta+e}2^{\lfloor 2^{\delta m}\rfloor}2^{cm}\Pr^y_x[D(x)=1]_\theta.$$

If the size of D is $\leq 2^{\epsilon m}-\lceil 2^{(\delta+\gamma)m}\rceil-2$, the same inequality holds for $\neg D$ instead of D.

3 Self-Provability via Feasible Witnessing

3.1 Self-Provability for Discrete Logarithm

We show that the random self-reducibility of the discrete logarithm problem can be used to derive a conditional self-provability of the statement that the discrete logarithm problem can be solved by p-size circuits.

For simplicity, we consider the discrete logarithm problem for $\mathbb{Z}_q^\times$, multiplicative groups of integers modulo a prime q. Let G be such a cyclic group. Then there are p-time algorithms A_1, A_2 such that $A_1(g, h, q) = g \cdot h \in G$, for $g, h \in G$, and $A_2(g, q) = g^{-1}$, for $g \in G$. That is, A_1 (given q) defines the multiplication of two elements in G and A_2 outputs the inverse of each $g \in G$.

The discrete logarithm problem for a cyclic group G generated by g is defined as follows. Given $b \in G$, we want to find a such that $g^a = b$. The discrete logarithm problem is 'random self-reducible': If we have a circuit C which solves the problem for a p-fraction of all $b \in G$, we can turn it efficiently into a randomized circuit C' which solves the problem on each $b \in G$ with probability $\geq p$. The circuit C' interprets its random bits r as $r \in [|G|]$. Then C' applies C on bg^r. Since bg^r is a uniformly random element of G, C succeeds in finding ℓ such that $g^\ell = bg^r$ with probability $\geq p$. Finally, C' outputs $\ell - r$, which is the correct answer with probability $\geq p$. In other words, the following implication holds

$$\Pr_{b \in G}[g^{C(b)} = b] \geq p \rightarrow \forall b \in G, \Pr_{r \in [|G|]}[g^{C'(b,r)} = b] \geq p, \tag{3.1}$$

where C' is generated from C and g by a p-time function.

We want to express (3.1) by a propositional formula. To do so, we approximate probabilities by a Nisan-Wigderson generator based on a hard function $f \in \mathsf{E}$. Fix a constant k and assume that $q \in (2^{n-1}, 2^n]$, $n \in \mathbb{N}$. Then, for each n^k-size circuit C, the predicate $g^{C(b)} = b$ with input b and the predicate $g^{C'(b,r)} = b$ with input r are computable respectively by circuits D_1 and D_2 with n inputs and size $poly(n)$. Circuits D_1, D_2 reject all inputs not in G, so in particular $\Pr_{r \in [|G|]}[g^{C'(b,r)} = b] \leq 2 \Pr_{r \in \{0,1\}^n}[D_2(r) = 1]$. Further, for each $\epsilon < 1$, there are constants c', c'' and $poly(n)$-time computable generator $NW_f : \{0,1\}^{c'\lceil \log n \rceil} \mapsto \{0,1\}^n$ such that if $f : \{0,1\}^{c''\lceil \log n \rceil} \mapsto \{0,1\}$ is hard to $(1/2 + 1/2^{\epsilon c'' \lceil \log n \rceil})$-approximate by circuits of size $2^{\epsilon c'' \lceil \log n \rceil}$, then

$$\left| \Pr_{z \in \{0,1\}^{c'\lceil \log n \rceil}}[D_1(NW_f(z)) = 1] - \Pr_{b \in \{0,1\}^n}[D_1(b) = 1] \right| \leq 1/n,$$

$$\left| \Pr_{z \in \{0,1\}^{c'\lceil \log n \rceil}}[D_2(NW_f(z)) = 1] - \Pr_{r \in \{0,1\}^n}[D_2(r) = 1] \right| \leq 1/n.$$

Therefore, if f is hard, we have

$$\Pr_{z\in\{0,1\}^{c'\lceil\log n\rceil}}[D_1(NW_f(z))=1]\geq p\rightarrow\forall b\in G,\Pr_{z\in\{0,1\}^{c'\lceil\log n\rceil}}[D_2(NW_f(z))=1]\geq\frac{p}{2}-\frac{3}{2n}. \tag{3.2}$$

The advantage of (3.2) is that it can be expressed by $poly(n)$-size tautologies $\mathsf{self}_n(p,b,C)$ with free variables for n^k-size circuits C, n-bit strings b and n-bit parameters p (among other extension variables). The tautologies have p-size proofs in some proof system which includes a p-size proof of the primality of q and a p-size proof of the fact that g is a generator of G. Here, we use the property that g generates G if and only if $g^{(q-1)/d}\not\equiv 1 \pmod{q}$ for every prime d dividing $q-1$. We can thus define a Cook-Reckhow propositional proof system P_ϵ as EF with the additional axioms which allow the system P_ϵ to derive any substitutional instance of $\mathsf{self}_n(p,b,C)$ and $\mathsf{tt}(f,2^{\epsilon n},1/2-1/2^{\epsilon n})$ in a single step of the proof, for each sufficiently big n.

Theorem 8 (Self-provability for the discrete logarithm)**.**
Let k be a constant. Assume that for some $\epsilon<1$ we have a Boolean function $f\in\mathsf{E}$ such that for each sufficiently big n, f is not $(1/2+1/2^{\epsilon n})$-approximable by any $2^{\epsilon n}$-size circuit. Let P_ϵ be the propositional proof system defined above. If there are n^k-size circuits solving the discrete logarithm problem for $\mathbb{Z}_q^\times$, where $q\in(2^{n-1},2^n]$, then there are p-size circuits D such that P_ϵ has p-size proofs of tautologies encoding the statement "$\forall b\in\mathbb{Z}_q^\times, g^{D(b)}=b$."

Proof. Given an n^k-size circuit B solving the discrete logarithm problem for $\mathbb{Z}_q^\times$, where $q\in(2^{n-1},2^n]$, P_ϵ can derive $\mathsf{self}_n(1/2-1/n,b,B)$. Since the assumption of $\mathsf{self}_n(1/2-1/n,b,B)$ is true and since it contains essentially no free variables, it can be proven efficiently in EF. (Essentially, in order to do so, it suffices to evaluate a $\mathsf{P/poly}$-predicate inside EF.) Consequently, P_ϵ proves efficiently $\forall b\in\mathbb{Z}_q^\times, g^{D(b)}=b$, for a suitable p-size circuit D obtained by simulating C' on all $z\in\{0,1\}^{c'\lceil\log n\rceil}$. □

Theorem 8 establishes a conditional equivalence between a circuit lower bound and a proof complexity lower bound (for propositional formulas which might not be tautological). We formulate it for a proof system P'_ϵ defined analogously as P_ϵ. The only difference is that instead of formulas $\mathsf{self}_n(p,b,C)$, system P'_ϵ uses more general formulas $\mathsf{self}^s_n(p,b,C)$ in which the circuit C is allowed to have arbitrary size $s\geq n$ (and the parameters of f are adjusted accordingly).

Corollary 2. *Assume that for some $\epsilon<1$ we have a Boolean function $f\in\mathsf{E}$ such that for each sufficiently big n, f is not $(1/2+1/2^{\epsilon n})$-approximable by any $2^{\epsilon n}$-size circuit. Let P'_ϵ be the propositional proof system defined above.*

Then, there are $poly(n)$-size circuits solving the discrete logarithm problem for group $\mathbb{Z}_q^\times$, where $q\in(2^{n-1},2^n]$, if and only if there are $poly(n)$-size circuits D such that P'_ϵ has p-size proofs of formulas encoding the statement "$\forall b\in\mathbb{Z}_q^\times, g^{D(b)}=b$."

Proof. The implication '→' follows by Theorem 8. The opposite implication follows from the soundness of P'_ϵ. □

Using the same argument as in Corollary 2, we derive an unconditional equivalence between hardness of Discrete Logarithm and super-polynomial P''_ϵ lower bounds for the circuit upper bound formulas in the statement of the Corollary. Here P''_ϵ is a proof system which is verifiable by polynomial-size circuits, and is defined the same way as P'_ϵ but using axioms $\mathsf{tt}(f_n, 2^{\epsilon n}, 1/2 - 1/2^{\epsilon n})$ for unconditionally hard functions f_n, which can be shown to exist by a standard counting argument.

Item 1 of Theorem 1 follows by setting $Q = P'_\epsilon$ and the set $F_{k,n}$ to be the circuit upper bound formulas for Discrete Logarithm for each kn^k-size circuit D. Item 2 follows by setting $Q = P''_\epsilon$ with $F_{k,n}$ the same as before.

3.2 Conditional Self-Provability for SAT

We prove Theorem 2 from the Introduction, restated here for convenience.

Theorem 9 (Circuit complexity from proof complexity & witnessing of $\mathsf{NP} \not\subseteq \mathsf{P/poly}$). *Let $k \geq 1$ be a constant.*

1. *Suppose that there is a p-time function f such that for each big enough n, $w^k_n(f)$ is a tautology. If $\mathsf{EF} + w^k(f)$ is not p-bounded, then $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^{\epsilon k}]$ for infinitely many n.*

2. *Suppose that there is a p-time function f such that for some n_0, $\mathsf{S}^1_2 \vdash W^k_{n_0}(f)$. If EF is not p-bounded, then $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^{\epsilon k}]$ for infinitely many n.*

In Items 1 and 2, $\epsilon > 0$ is a universal constant (independent of k).

Proof. First we show Item 1.

Suppose that $\forall n > 1, \mathsf{SAT}_n \in \mathsf{Circuit}[n^{k'}]$, where $\mathsf{Circuit}[s(n)]$ stands for the set of all single-output circuits with n inputs and size $\leq s(n)$. Then, there is a sequence of circuits C with n inputs, $\leq n$ outputs and size n^k falsifying the right disjunct in $w^k_n(f)$, for some $k > k'$ and all $n > 1$. Therefore, $\mathsf{EF} + w^k(f)$ admits p-size proofs of $\mathsf{SAT}_n(x, y) \rightarrow \mathsf{SAT}_n(x, C(x))$. That is, the mere validity of $\mathsf{SAT}_n \in \mathsf{Circuit}[n^{k'}]$ implies an efficient propositional provability of $\mathsf{SAT}_n \in \mathsf{P/poly}$. The efficient provability of $\mathsf{SAT}_n(x, y) \rightarrow \mathsf{SAT}_n(x, C(x))$ further implies that $\mathsf{EF} + w^k(f)$ is p-bounded: To prove a tautology ϕ of size n in $\mathsf{EF} + w^k(f)$ it suffices to check out that $\neg\mathsf{SAT}_n(\neg\phi, C(\neg\phi))$ (which implies that $\mathsf{SAT}_n(\phi, y)$ and ϕ hold).

To see Item 2, note that by the correspondence between S^1_2 and EF, cf. [38], if $W^k_{n_0}(f)$ was provable in S^1_2, for some p-time f, then tautologies $w^k_n(f)$, for $n \geq n_0$, would have p-size proofs in EF.

□

Restricting nonuniformity. In Theorem 2, we can restrict the number of nonuniform bits in the concluded lower bounds by adapting formulas $w_n^k(f)$: Assume that the circuit C includes a hardwired description of a fixed universal Turing machine U. Moreover, interpret C as encoding an algorithm A described by $\leq \log n$ bits with $u(n) \leq n^k$ nonuniform bits of advice $a(n)$. The algorithm A and its nonuniform advice are described by free variables. We assume that u is p-time. On each input $z \in \{0,1\}^n$, $C(z)$ uses U to simulate the computation of A on z with access to $a(n)$ up to n^k steps. That is, now the size of C is $poly(n^k)$. Denote the resulting formulas by $w_n^{k,u}(f)$. If we have a p-time function f which witnesses errors of n^k-time algorithms described by $\log n$ bits with $u(n)$ bits of advice attempting to solve the search version of SAT, i.e. such that formulas $w_n^{k,u}(f)$ are tautologies for big enough n, we can define proof system $\mathsf{EF} + w^{k,u}(f)$. Further, we can define $\forall\Pi_1^b$ formulas $W_n^{k,u}(f)$ expressing "$\forall n > n_0$, $w_n^{k,u}(f)$." Denote by $\mathsf{Time}[n^k]/u(n)$ the class of problems solvable by uniform algorithms with $\leq u(n)$ bits of nonuniform advice running in time $O(n^k)$. The proof of Theorem 2 works in this case as well.

Corollary 3 (Circuit complexity from proof complexity & witnessing of $\mathsf{P} \neq \mathsf{NP}$)**.**
Let $k \geq 1$ be a constant and u a p-time function such that $u(n) \leq n^k$.

1. *Suppose that there is a p-time function f such that for each big enough n, $w_n^{k,u}(f)$ is a tautology. If $\mathsf{EF} + w^{k,u}(f)$ is not p-bounded, then $\mathsf{SAT} \notin \mathsf{Time}[n^{\Omega(k)}]/u(n)$.*

2. *Suppose that there is a p-time function f such that for some n_0, $\mathsf{S}_2^1 \vdash W_{n_0}^{k,u}(f)$. If EF is not p-bounded, then $\mathsf{SAT} \notin \mathsf{Time}[n^{\Omega(k)}]/u(n)$.*

The significance of Corollary 3 is that in the uniform setting, a similar kind of feasible witnessing is known to exist using diagonalization techniques [25, 10]. It is unclear whether diagonalization techniques will suffice to establish that $w_n^{k,u}(f)$ is a tautology for large enough n within some concrete proof system, but there is at least a strong motivation for considering the question, given its implications for deriving strong computational complexity lower bounds from proof complexity lower bounds.

4 Feasible anticheckers.

If there is an n^k-size circuit computing SAT_n, there is a $poly(n^k)$-size circuit B with n inputs and $\leq n$ outputs such that $\forall x, y \in \{0,1\}^n, (\mathsf{SAT}_n(x,y) \to \mathsf{SAT}_n(x, B(x)))$. We use this to formulate the existence of anticheckers for SAT as a $\forall\Pi_1^b$ statement.

Theorem 10 ('CC ← PC' from feasible anticheckers)**.**
Let $k \geq 3$ be a constant and assume that there is a p-time function f such that S_2^1 proves:

"*$\forall 1^n$, $f(1^n)$ is a $poly(n^k)$-size circuit B such that*

$$\forall x, y \in \{0,1\}^n, [\mathsf{SAT}_n(x,y) \to \mathsf{SAT}_n(x, B(x))]$$

or $\big(f(1^n)$ *outputs sets* $A_n^{\mathsf{SAT}_n,n^k}, A' \subseteq \{0,1\}^n$*,* $D \subseteq A_n^{\mathsf{SAT}_n,n^k} \times A'$ *of size* $poly(n^k)$ *such that*

$$\forall x \in A_n^{\mathsf{SAT}_n,n^k}[\exists y_x \in A', \langle x, y_x\rangle \in D \wedge \forall z, y \in A', (\langle x,y\rangle \in D \wedge \langle x,z\rangle \in D \to y = z)]$$

and $\forall$ n^k*-size circuit* C*,*

$$\forall x \in A_n^{\mathsf{SAT}_n,n^k} \forall y \in \{0,1\}^n\ [\mathsf{SAT}_n(x,y) \to \mathsf{SAT}_n(x,y_x)] \wedge$$

$$\exists x \in A_n^{\mathsf{SAT}_n,n^k}, \mathsf{SAT}_n(x,y_x) \neq C(x)\big)\text{.''}$$

Then, proving that EF *is not p-bounded implies* $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^k]$ *for infinitely many* n*.*

Proof. The statement assumed to have an S^1_2-proof is $\forall\Pi^b_1$, so there are p-size EF-proofs of its propositional translation. If we now assume that $\exists n_0 \forall n > n_0, \mathsf{SAT}_n \in \mathsf{Circuit}[n^k]$, there are circuits C and $y \in \{0,1\}^n$ falsifying the second disjunct of the translated assumption for $n > n_0$. Consequently, EF proves efficiently that the circuits generated by $f(1^n)$ solve SAT_n, which implies that EF is p-bounded. □

Existential quantifiers instead of witnessing. If we used the existential quantifiers instead of function f in Theorem 10, the resulting statement S formalizing the existence of anticheckers would be $\forall\Sigma^b_2$. By the KPT theorem [39], PV_1-provability of S would then imply the existence of p-time functions $f_1, \dots, f_c$, for a constant c, with a PV_1-proof of:

> "$\forall 1^n$, $\forall x^1, \dots, x^c, y^1, \dots, y^c, \tilde{y}^1, \dots, \tilde{y}^c \in \{0,1\}^n, \forall n^k$-size circuits $C^1, \dots, C^c$, $f_1(1^n)$ outputs a $poly(n^k)$-size circuit B and $A_n^{\mathsf{SAT}_n,n^k}, A' \subseteq \{0,1\}^n, D \subseteq A_n^{\mathsf{SAT}_n,n^k} \times A'$ of size $poly(n^k)$ such that the following predicate, denoted $P_{f_1}(x^1, y^1, C^1, \tilde{y}^1)$, holds:
>
> $$\big(\mathsf{SAT}_n(x^1,y^1) \to \mathsf{SAT}_n(x^1, B(x^1))\big) \vee \big(D'(\tilde{y}^1) \wedge \exists \tilde{x} \in A_n^{\mathsf{SAT}_n,n^k}, \mathsf{SAT}_n(\tilde{x}, y_{\tilde{x}}) \neq C(\tilde{x})\big),$$
>
> where $D'(\tilde{y}^1)$ stands for the remaining part of the Σ^b_0 subformula of S,
> or $f_2(1^n, x^1, y^1, C^1, \tilde{y}^1)$ outputs a $poly(n^k)$-size circuit B and $A_n^{\mathsf{SAT}_n,n^k}, A', D$ of size $poly(n^k)$ such that $P_{f_2}(x^2, y^2, C^2, \tilde{y}^2)$ holds, or
> ...
> or $f_c(1^n, x^1, \dots, x^{c-1}, y^1, \dots, y^{c-1}, C^1, \dots, C^{c-1}, \tilde{y}^1, \dots, \tilde{y}^{c-1})$ outputs a $poly(n^k)$-size circuit B and $A_n^{\mathsf{SAT}_n,n^k}, A', D$ of size $poly(n^k)$ such that $P_{f_c}(x^c, y^c, C^c, \tilde{y}^c)$."

The resulting $\forall\Pi^b_1$-statement could be translated to propositional tautologies with p-size EF-proofs. However, given $\forall n, \mathsf{SAT}_n \in \mathsf{Circuit}[n^k]$, we could not directly obtain p-size EF-proofs of tautologies stating that one of the functions $f_1, \dots, f_c$ generates a circuit solving SAT_n. This is because B and $A_n^{\mathsf{SAT}_n,n^k}$ generated by f_2 depend on y^1. For the same reason, it seems possible for EF to prove efficiently $\mathsf{SAT} \in \mathsf{P/poly}$ (using the formalization based on the KPT witnessing) without proving efficiently all tautologies.

5 One-way functions from $\mathsf{NP} \not\subseteq \mathsf{P/poly}$

Theorem 11 ('CC ← PC' from 'OWF ← $\mathsf{NP} \not\subseteq \mathsf{P/poly}$' & hardness of E)**.**
Assume that for each sufficiently big n, each $2^{n/4}$-size circuit fails to compute $h' \in \mathsf{E}$ on $\geq (1/2 - 1/2^{n/4})$ of all inputs. Further, assume that there is a p-time function $h : \{0,1\}^n \mapsto \{0,1\}^{u(n)}$ such that for each constants c, d, there is a p-time function f_2 and a constant $0 < \epsilon < 1$ such that S^1_2 proves:

> *"$\forall n, \forall$ cn^c-size circuit C with $u(m)$ inputs and m outputs such that $n \leq dm^d$, $\big(f_2(C)$ is a poly(n)-size circuit B such that*
>
> $$\forall x, y \in \{0,1\}^{\lfloor n^\epsilon \rfloor}, [\mathsf{SAT}_{\lfloor n^\epsilon \rfloor}(x, y) \rightarrow \mathsf{SAT}_{\lfloor n^\epsilon \rfloor}(x, B(x))]$$
>
> *or*
>
> $$\overset{y}{\Pr_{x \in \{0,1\}^m}}[h(C(h(x))) = h(x)]_{\frac{1}{m}} < 1/2\big)."$$

Then, proving that $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ is not p-bounded implies $\mathsf{SAT} \notin \mathsf{P/poly}$.

The system $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ is defined in the same way as in the introduction. That is, $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ is an extension of EF which can use substitutional instances of $\mathsf{tt}(h'_n, 2^{n/4}, 1/2 - 1/2^{n/4})$, for sufficiently big n, in its proofs.

The proof of Theorem 11 is based on the following lemma formalizing a conditional witnessing of $\mathsf{NP} \not\subseteq \mathsf{P/poly}$, cf. [46, 44].

Lemma 2 (Formalized witnessing of $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ from OWF & hardness of E)**.**
Let $k \geq 1$ be a constant. For each p-time functions $h : \{0,1\}^n \mapsto \{0,1\}^{u(n)}$ and f_1, there are p-time functions f_0, f_{-1}, f_{-2} and constants b, n_1 such that $\mathsf{S}^1_2 + dWPHP(\mathsf{PV})$ proves: "$\forall 1^n > n_1$, $\forall m$ such that $n/2^b \leq 2^{bm} \leq n$, if

$$\mathsf{LB_{tt}}'(f_1(1^{2^m}), 2^{m/4}, 2^m(1/2 - 1/2^{m/4})),$$

> *then $f_0(1^n, m)$ outputs sets $A, A' \subseteq \{0,1\}^n$ of size poly(n) such that*
>
> $$\forall x \in A\ \exists y_x \in A'\ \mathsf{SAT}_n(x, y_x)$$

and $\big(\forall$ n^k-size circuit C with n inputs and $\leq n$ outputs,

$$\exists x \in A, \neg\mathsf{SAT}_n(x, C(x))$$

or $f_{-1}(C, m)$ outputs a poly(n)-size circuit C' with $u(n')$ inputs and n' outputs, where $n \leq f_{-2}(1^{n'})$, such that

$$\overset{y}{\Pr_{x \in \{0,1\}^{n'}}}[h(C'(h(x))) = h(x)]_{\frac{1}{n'}} \geq 1/2\big)."$$

Here, $\mathsf{LB_{tt}}'$ is obtained from $\mathsf{LB_{tt}}$ by setting $m_0 = 0$ and skipping the universal quantifier on m (so m in $\mathsf{LB_{tt}}'$ is the same as the universally quantified m in the S^1_2-provable statement).

Proof of Theorem 11 from Lemma 2. Intuitively, Theorem 11 assumes that the hardness of SAT yields a function h which is hard to invert. Lemma 2 shows that such h can be used to find an error of each small circuit attempting to compute SAT. Combining the assumption of Theorem 11 with Lemma 2 we obtain a p-time function f such that for each small circuit C, either C solves SAT or $f(C)$ finds an error of C. Moreover, this holds provably in $\mathsf{S}^1_2 + dWPHP(\mathsf{PV})$, so the propositional translation of the correctness of the witnessing statement has short proofs in $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$. This will allow us to derive the desired implication similarly as in the proof of Theorem 2.

We proceed with a formal proof.

The assumption of Theorem 11 in combination with Lemma 2 implies that for each $k \geq 1$, for p-time f_1 generating the truth-table of h', there is $0 < \epsilon < 1$ and b, n_1 such that $\mathsf{S}^1_2 + dWPHP(\mathsf{PV})$ proves the following statement S:

"$\forall 1^n > n_1$, $\forall m, n/2^b \leq 2^{bm} \leq n$, if

$$\mathsf{LB}_{\mathsf{tt}}{}'(f_1(1^{2^m}), 2^{m/4}, 2^m(1/2 - 1/2^{m/4})),$$

then $f_0(1^n, m)$ outputs $A, A' \subseteq \{0,1\}^n$ such that

$$\forall x \in A\ \exists y_x \in A'\ \mathsf{SAT}_n(x, y_x)$$

and $\big(\forall n^k$-size circuit C with n inputs and $\leq n$ outputs,

$$\exists x \in A, \neg \mathsf{SAT}_n(x, C(x))$$

or $f_2(f_{-1}(C, m))$ outputs a circuit B such that

$$\forall x, y \in \{0,1\}^{\lfloor n^\epsilon \rfloor}, [\mathsf{SAT}_{\lfloor n^\epsilon \rfloor}(x, y) \rightarrow \mathsf{SAT}_{\lfloor n^\epsilon \rfloor}(x, B(x))]$$

or y' does not satisfy the assumption of $\Pr_x^{y'}[\cdot]_{1/n'} \geq 1/2\big)$."

Since S is $\forall \Sigma^b_1$, by Lemma 1, there is a p-time function f_3 and a constant ℓ such that PV_1 proves: "$\forall 1^n > n_1, \forall m, n/2^b \leq 2^{bm} \leq n$, if $|h'| \geq n^\ell$ and a $2^{||h'||/4}$-size circuit generated by a p-time function fails to compute h', then $f_3(1^n, m, h', C, x, y, y')$ outputs a circuit falsifying

$$\mathsf{LB}_{\mathsf{tt}}{}'(f_1(1^{2^m}), 2^{m/4}, 2^m(1/2 - 1/2^{m/4})),$$

or $f_3(1^n, m, h', C, x, y, y')$ outputs a circuit falsifying the assumption of $\Pr_x^{y'}[\cdot]_{1/n'} \geq 1/2$ or F' holds," where F' is the rest of the statement S.

Consequently, EF proves efficiently the propositional translation of the PV_1-theorem. Substituting h' for y', $\mathsf{EF}^+ := \mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ proves the formula F encoding F'. We now proceed as in the proof of Theorem 2. Assuming that $\mathsf{SAT} \in \mathsf{P/poly}$, there is some k such that for all $1^n > n_1 \geq 1$ we can efficiently falsify the first disjunct of F. Therefore, there is a p-size circuit B such that EF^+ proves efficiently $\mathsf{SAT}_{\lfloor n^\epsilon \rfloor}(x, y) \rightarrow \mathsf{SAT}_{\lfloor n^\epsilon \rfloor}(x, B(x))$, which means that EF^+ is p-bounded. □

Proof of Lemma 2. Let $f_0(1^n, m)$ output the set of propositional formulas $A := \{\phi_z(x) \mid z \in \{0,1\}^{cm}\}$, where $\phi_z(x)$ uses free variables x together with some auxiliary variables and encodes the statement

$$h(x) = h(NW_f(z)).$$

Here, $NW_f : \{0,1\}^{cm} \mapsto \{0,1\}^{\lfloor 2^{\delta m} \rfloor}$ and c are given by Theorem 7, for some $0 < \gamma, \delta < 1$ specified later, and $f = f_1(1^{2^m})$. The size of $\phi_z(x)$ is $\lfloor 2^{K\delta m} \rfloor$, for a constant K depending only on h. We set m so that $n/2^{K\delta} \leq 2^{K\delta m} \leq n$ and treat formulas $\phi_z(x)$ as formulas of size n. Let $A' := \{NW_f(z) \mid z \in \{0,1\}^{cm}\}$.

We reason in $\mathsf{S}^1_2 + dWPHP(\mathsf{PV})$. Suppose that an n^k-size circuit C with n inputs and $\leq n$ outputs finds a satisfying assignment for all formulas in A. Then, there is an n^k-size circuit C' with $u(\lfloor 2^{\delta m} \rfloor)$ inputs such that

$$2^{\lfloor 2^{\delta m} \rfloor} \times \{z \in \{0,1\}^{cm} \mid h(C'(h(NW_f(z)))) \neq h(NW_f(z))\} \preceq_0 0. \tag{5.1}$$

The circuit C' is obtained from C by a p-time algorithm f_{-1} depending on δ, h and m. The predicate $h(C'(h(x))) \neq h(x)$ is computable by a $2^{K'\delta m}$-size circuit D with $\lfloor 2^{\delta m} \rfloor$ inputs, for a constant K' depending only on k and h. Now, we set a sufficiently small γ and δ so that $2^{K'\delta m} \leq 2^{m/4} - \lceil 2^{(\delta+\gamma)m} \rceil - 1$. Therefore, by Theorem 7, the assumption that f is hard on average for $2^{m/4}$-size circuits implies that

$$2^{\lfloor 2^{\delta m} \rfloor} \times \{z < 2^{cm} \mid D(NW_f(z)) = 1\} \succeq_e 2^{cm} \times \{x < 2^{\lfloor 2^{\delta m} \rfloor} \mid D(x) = 1\},$$

where $e = \lceil 2^{\delta m} \rceil / 2^{m/4}$. Consequently, by (5.1) and Proposition 2 *ii*),

$$0 \succeq_e 2^{cm} \times \{x < 2^{\lfloor 2^{\delta m} \rfloor} \mid h(C'(h(x))) \neq h(x)\}. \tag{5.2}$$

We want to show that $\Pr^y_{x \in \{0,1\}^{\lfloor 2^{\delta m} \rfloor}}[h(C'(h(x))) = h(x)]_{\frac{1}{\lfloor 2^{\delta m} \rfloor}} \geq 1/2$. For the sake of contradiction, assume that this is not the case. Then, $\{x < 2^{\lfloor 2^{\delta m} \rfloor} \mid h(C'(h(x)) = h(x)\} \preceq_{1/\lfloor 2^{\delta m} \rfloor} 2^{\lfloor 2^{\delta m} \rfloor - 1}$. By Item 1 *iii*) of Proposition 3,

$$\{x < 2^{\lfloor 2^{\delta m} \rfloor} \mid h(C'(h(x)) \neq h(x)\} \succeq_{2/\lfloor 2^{\delta m} \rfloor} 2^{\lfloor 2^{\delta m} \rfloor - 1}.$$

By Proposition 2 *iii*),

$$2^{cm} \times \{x < 2^{\lfloor 2^{\delta m} \rfloor} \mid h(C'(h(x)) \neq h(x)\} \succeq_{2/\lfloor 2^{\delta m} \rfloor} 2^{\lfloor 2^{\delta m} \rfloor - 1 + cm}. \tag{5.3}$$

By Proposition 2 *ii*), (5.2) and (5.3) yield $0 \succeq_{2/\lfloor 2^{\delta m} \rfloor + e} 2^{\lfloor 2^{\delta m} \rfloor - 1 + cm}$. Hence, by Item 1 *ii*) of Proposition 3, $2^{\lfloor 2^{\delta m} \rfloor - 1 + cm} < (3/\lfloor 2^{\delta m} \rfloor + e) 2^{\lfloor 2^{\delta m} \rfloor - 1 + cm}$, which is a contradiction for a sufficiently big m. □

6 Learning from the non-existence of OWFs

Theorem 12 (‘CC $\leftarrow$ PC’ from ‘Learning $\leftarrow \nexists$ OWF’ & hardness of E)**.**
Let $k, t \geq 1$ be constants. Assume that for each sufficiently big n, each $2^{n/4}$-size circuit fails to compute $h' \in \mathsf{E}$ on $\geq 1/2 - 1/2^{n/4}$ of all inputs. Further, assume that there is a p-time function $h : \{0,1\}^n \mapsto \{0,1\}^{u(n)}$ such that for each constants c, d, there is a p-time function f_2 and constants n_0 and $0 < \epsilon < 1$ such that S^1_2 proves:

> *“$\forall n$, $\forall$ cn^c-size circuit C with $u(m)$ inputs and m outputs such that $n \leq dm^d$, $\big(f_2(C)$ outputs a poly(n)-size circuit B learning $\lfloor n^\epsilon \rfloor^t$-size circuits with $\lfloor n^\epsilon \rfloor$ inputs over the uniform distribution, up to error $1/2 - 1/\lfloor n^\epsilon \rfloor$, with confidence $1/\lfloor n^\epsilon \rfloor$; formally, $\forall f : \{0,1\}^{\lfloor n^\epsilon \rfloor} \mapsto \{0,1\}$, $\forall$ $\lfloor n^\epsilon \rfloor^t$-size circuit D computing f,*
>
> $$\overset{y}{\underset{w}{\Pr}}[B(1^{\lfloor n^\epsilon \rfloor}, w)\ (1/2 + 1/\lfloor n^\epsilon \rfloor)\text{-}\mathit{approximates}\ f]_{1/2\lfloor n^\epsilon \rfloor} \geq 1/\lfloor n^\epsilon \rfloor;$$
>
> *or*
>
> $$\overset{y}{\underset{x \in \{0,1\}^m}{\Pr}}[h(C(h(x))) = h(x)]_{\frac{1}{m}} < 1/2\big).\text{”}$$

Then there are constants b and a (depending on $k, t, h, h', c, d, f_2, n_0, \epsilon$) such that for each n the existence of a function $g_n : \{0,1\}^n \mapsto \{0,1\}$ such that no circuit of size bn^b computes g_n on $(1/2 + 1/n)$ fraction of inputs and such that $\mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ does not have 2^{an}-size proofs of $\mathsf{tt}(g_n, n^t)$ implies that $\mathsf{SAT}_n \notin \mathsf{Circuit}[n^k]$.

Note that the S^1_2-theorem in the assumption of Theorem 12 assumes $2^{\lfloor n^\epsilon \rfloor} \in \mathit{Log}$.

Proof. The assumption of Theorem 12 in combination with Lemma 2 implies that for any given $k, t \geq 1$, for p-time f_1 generating the truth-table of h', there are constants $0 < \epsilon < 1$ and b, n_1 such that $\mathsf{S}^1_2 + dWPHP(\mathsf{PV})$ proves the following statement S:

“$\forall 1^n > n_1, \forall m, n/2^b \leq 2^{bm} \leq n$, if

$$\mathsf{LB_{tt}}'(f_1(1^{2^m}), 2^{m/4}, 2^m(1/2 - 1/2^{m/4})),$$

then $f_0(1^n, m)$ outputs $A, A' \subseteq \{0,1\}^n$ such that

$$\forall x \in A\ \exists y_x \in A'\ \mathsf{SAT}_n(x, y_x)$$

and $\big(\forall n^k$-size circuit C with n inputs and $\leq n$ outputs,

$$\exists x \in A, \neg\mathsf{SAT}_n(x, C(x))$$

or $f_2(f_{-1}(C,m))$ outputs a circuit B such that $\forall f:\{0,1\}^{\lfloor n^\epsilon \rfloor} \mapsto \{0,1\}$, $\forall$ $\lfloor n^\epsilon \rfloor^t$-size circuit D computing f,

$$\Pr_w^y[B(1^{\lfloor n^\epsilon \rfloor}, w)\ (1/2+1/\lfloor n^\epsilon \rfloor)\text{-}\mathit{approximates}\ f]_{1/2\lfloor n^\epsilon \rfloor} \geq 1/\lfloor n^\epsilon \rfloor,$$

or y' does not satisfy the assumption of $\Pr_x^{y'}[\cdot]_{1/n'} \geq 1/2\big)$."

Since S is $\forall\Sigma_1^b$, by Lemma 1, there is a p-time function f_3 and a constant ℓ such that PV_1 proves: "$\forall 1^n > n_1, \forall m, n/2^b \leq 2^{bm} \leq n$, if $|h'| \geq 2^{\ell \lfloor n^\epsilon \rfloor}$ and a $2^{||h'||/4}$-size circuit generated by a p-time function fails to compute h', then $f_3(1^n, m, h', C, f, D, y, y')$ outputs a circuit falsifying

$$\mathsf{LB_{tt}}'(f_1(1^{2^m}), 2^{m/4}, 2^m(1/2-1/2^{m/4})),$$

or $f_3(1^n, m, h', C, f, D, y, y')$ outputs a circuit falsifying the assumption of $\Pr_x^{y'}[\cdot]_{1/n'} \geq 1/2$ or it outputs a circuit falsifying the assumption of $\Pr_w^y[\cdot]_{1/2\lfloor n^\epsilon \rfloor} \geq 1/\lfloor n^\epsilon \rfloor$ or F' holds," where F' is the rest of the statement S.

Consequently, $\mathsf{EF}^+ := \mathsf{EF} + \mathsf{tt}(h', 2^{n/4}, 1/2 - 1/2^{n/4})$ proves efficiently the propositional translation of F'. If we now fix $n > 1$ and assume that $\mathsf{SAT}_n \in \mathsf{Circuit}[n^{k''}]$, then there is some $k = O(k'')$ such that we can efficiently falsify the first disjunct of the propositional translation of F' in EF^+. Therefore, there is a $\mathit{poly}(n)$-size circuit B and a 2^{Kn}-size EF^+ proof of $\mathsf{lear}^{h'}_{1/2n}(B, \mathsf{Circuit}[n^t], 1/2 - 1/n, 1/n)$, for a constant K independent of n. Recall that this means that EF^+ proves efficiently $\neg\mathsf{tt}(f, n^t) \rightarrow R$, for a formula R.

Let $b \geq t$ be such that B has size $\leq bn^b$. We claim that for each Boolean function $g_n : \{0,1\}^n \mapsto \{0,1\}^n$ which is not $(1/2+1/n)$-approximable by any circuit of size bn^b, there is a 2^{an}-size EF^+-proof of $\mathsf{tt}(g_n, n^t)$, for a constant a independent of n. This is because in order to prove $\mathsf{tt}(g_n, n^t)$ in EF^+, it suffices to check in EF^+ that $\neg R$ holds for $f = g_n$. $\neg R$ holds for $f = g_n$ as otherwise there would be a bn^b-size circuit $(1/2+1/n)$-approximating g_n. Moreover, the fact that $\neg R$ holds for $f = g_n$ is efficiently provable in EF^+ as w.l.o.g. $\neg R$, for $f = g_n$, does not contain any free variables (we can assume that the auxiliary variables are substituted by suitable constants). □

References

[1] Aaronson S.; *Is P Versus NP Formally Independent?*; Bulletin of EATCS, 81: 109-136, 2003.

[2] Akavia A., Goldreich O., Goldwasser S., Moshkovitz D.; *On basing one-way functions on NP-Hardness*; STOC, 2006.

[3] Althöfer I.; *On sparse approximations to randomized strategies and convex combinations*; Linear Algebra and its Applications, 199(1):339-355, 1994.

[4] Ajtai M.; *The Complexity of the Pigeonhole Principle*; Combinatorica 14(4): 417-433, 1994.

[5] Atserias A.; *Distinguishing SAT from polynomial-size circuits, through black-box queries*; Computational Complexity Conference (CCC), 2006.

[6] Beame P., Pitassi T.; *Propositional Proof Complexity: Past, Present and Future*; Current Trends in Theoretical Computer Science, Entering the 21st Century, 42-70, 2001.

[7] Beyersdorff O., Bonacina I., Chew L.; Lower Bounds: From Circuits to QBF Proof Systems; ITCS, 2016.

[8] Beyersdorff O., Pich J.; Understanding Gentzen and Frege Systems for QBF; LICS, 2016.

[9] Binnendyk E., Carmosino M., Kolokolova A., Ramyaa R., Sabin M.; *Learning with distributional inverters*; Algorithmic Learning Theory (ALT), 2022.

[10] Bogdanov A., Talwar K., Wan A.; *Hard instances for satisfiability and quasi-one-way functions*; ICS, 2010.

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

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

[13] Buss S.; *Bounded arithmetic*; Bibliopolis, 1986.

[14] Beame P., Impagliazzo R.; Pitassi T.; *Exponential Lower Bounds for the Pigeonhole Principle*; Computational Complexity, 3:97-140, 1993.

[15] Bogdanov A., Brzuska; *On Basing Size-Verifiable One-Way Functions on NP-Hardness*; TCC (1), 2015.

[16] Bogdanov A., Trevisan L.; *On Worst-Case to Average-Case Reductions for NP Problems*; SIAM Journal on Computing, 36(4): 1119-1159, 2006.

[17] Chen L., Jin C., Santhanam R., Williams R.; *Constructive Separations and Their Consequences*; FOCS, 2021.

[18] Cobham A.; *The intrinsic computational difficulty of functions*; Proceedings of the 2nd International Congress of Logic, Methodology and Philosophy of Science, North Holland, pp. 24-30, 1965.

[19] Cook S.A.; *Feasibly constructive proofs and the propositional calculus*; Symposium on Theory of Computing (STOC), 1975.

[20] Cook S.A., Thapen N.; *The strength of replacement in weak arithmetic*; ACM Transactions on Computational Logic, 7(4):749-764, 2006.

[21] Cook S.A., Krajíček J.; *Consequences of the Provability of NP⊆P/poly*; Journal of Symbolic Logic, 72:1353-1357, 2007.

[22] Cook S.A., Reckhow R.; *The Relative Efficiency of Propositional Proof Systems*; Journal of Symbolic Logic, 44(1):36-50, 1979.

[23] de Rezende S., Göös M., Robere R.; *Proofs, Circuits and Communication*; SIGACT News Complexity Theory Column, 2022.

[24] Grochow J., Pitassi T.; *Circuit Complexity, Proof Complexity, and Polynomial Identity Testing: The Ideal Proof System*; Journal of the ACM, 65(6), 37:1-59, 2018.

[25] Gutfreund D., Shaltiel R., Ta-Shma A.; *If NP languages are hard in the worst-case then it is easy to find their hard instances*; Computational Complexity, 16(4), 412-441, 2007.

[26] Haken A.; *The Intractability of Resolution*; Theoretical Computer Science, 39: 297-308, 1985.

[27] Hirahara S.; *Non-black-box worst-case to average-case reductions within* NP; Foundations of Computer Science (FOCS), 2018.

[28] Impagliazzo R.; *A personal view of average-case complexity*; Structure in Complexity Theory (SCT), 1995.

[29] Impagliazzo R., Kabanets V., Wigderson A.; *In search of an easy witness: exponential time vs. probabilistic polynomial time*; J.Comp.Syst.Sci., 65(4), 672-694, 2002.

[30] Jeřábek E.; *Dual weak pigeonhole principle, Boolean complexity and derandomization*; Annals of Pure and Applied Logic, 129:1-37, 2004.

[31] Jeřábek E.; *Weak pigeonhole principle and randomized computation*; Ph.D. thesis, Charles University in Prague, 2005.

[32] Jeřábek E.; *Approximate counting in bounded arithmetic*; Journal of Symbolic Logic, 72:959-993, 2007.

[33] Kabanets V.; *Easiness Assumptions and Hardness Tests: Trading Time for Zero Error*; Journal of Computer and System Sciences, 63(2): 236-252, 2001.

[34] Krajíček J.; *Bounded arithmetic, propositional logic, and complexity theory*; Cambridge University Press, 1995.

[35] Krajíček J.; *Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic*; Journal of Symbolic Logic, 66(2):457-486, 1997.

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

[37] Krajíček J.; *Forcing with random variables and proof complexity*; Cambridge University Press, 2011.

[38] Krajíček J.; *Proof complexity*; Cambridge University Press, 2019.

[39] Krajíček J., Pudlák P., Takeuti G.; *Bounded arithmetic and the polynomial hierarchy*; Annals of Pure and Applied Logic, 52:143-153, 1991.

[40] Krajíček J., Pudlák P., Woods A.; *An Exponential Lower Bound to the Size of Bounded Depth Frege Proofs of the Pigeonhole Principle*; Random Structures and Algorithms 7(1): 15-40, 1995.

[41] Li, F., Tzameret I., Wang, Z.; *Characterizing Propositional Proofs as Noncommutative Formulas*; SIAM Journal on Computing 47(4): 1424-1462, 2018.

[42] Lipton R.J., Young N.E.; *Simple strategies for large zero-sum games with applications to complexity theory*; Symposium on Theory of Computing (STOC), 1994.

[43] Liu Y., Pass R.; *On one-way functions from NP-complete problems*; Computational Complexity Conference (CCC), 2022.

[44] Müller M., Pich J.; *Feasibly constructive proofs of succinct weak circuit lower bounds*; Annals of Pure and Applied Logic, 2019.

[45] Newman I.; *Private vs common random bits in communication complexity*; Information Processing Letters, 39:67-71, 1991.

[46] Pich J.; *Circuit lower bounds in bounded arithmetics*; Annals of Pure and Applied Logic, 166(1):29-45, 2015.

[47] Pich J., Santhanam R.; *Learning algorithms versus automatability of Frege systems*; arXiv, 2021.

[48] Pich J., Santhanam R.; *Why are Proof Complexity Lower Bounds Hard?*; FOCS, 2019.

[49] Pitassi T., Tzameret I.; *Algebraic proof complexity: progress, frontiers and challenges*; ACM SIGLOG News 3(3): 21-43, 2016

[50] Razborov A.A.; *Bounded arithmetic and lower bounds in boolean complexity*; Feasible Mathematics II, 344-386, 1995.

[51] Razborov A.A.; *Pseudorandom generators hard for k-DNF Resolution and Polynomial Calculus*; Annals of Mathematics, 181(2):415-472, 2015.

[52] Santhanam R.; *Pseudorandomness and the Minimum Circuit Size Problem*; Innovations in Theoretical Computer Science (ITCS), 2020.

[53] Santhanam R., Tzameret I.; *Iterated lower bound formulas: a diagonalization-based approach to proof complexity*; STOC, 2021.

[54] Williams R.; *Non-uniform ACC circuit lower bounds*; Computational Complexity Conference (CCC), 2011.