Electronic Colloquium on Computational Complexity
ECCC Technical Reports Series 1994 Report Nr: TR94–010

Natural Proofs

Alexander A. Razborov[†]
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540
and
Steklov Mathematical Institute
Vavilova 42, 117966, GSP–1
Moscow, RUSSIA

Steven Rudich[‡]
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15212

Received *December 13, 1994*

Abstract. We introduce the notion of *natural* proof. We argue that the known proofs of lower bounds on the complexity of explicit Boolean functions in non-monotone models fall within our definition of natural. We show based on a hardness assumption that natural proofs can't prove superpolynomial lower bounds for general circuits. Without the hardness assumption, we are able to show that they can't prove exponential lower bounds (for general circuits) applicable to the discrete logarithm problem. We show that the weaker class of AC^0-natural proofs which is sufficient to prove the parity lower bounds of Furst, Saxe, and Sipser; Yao; and Hastad is inherently incapable of proving the bounds of Razborov and Smolensky. We give some formal evidence that natural proofs are indeed natural by showing that every formal complexity measure which can prove superpolynomial lower bounds for a single function, can do so for almost all functions, which is one of the key requirements to a natural proof in our sense.

Keywords: lower bounds in non-uniform complexity, combinatorial properties, pseudo-random generators

[†] Supported by the grant # 93-6-6 of the Alfred P. Sloan Foundation, by the grant # 93-011-16015 of the Russian Foundation for Fundamental Research, and by an AMS-FSU grant
[‡] Partially supported by NSF grant CCR-9119319

1. Introduction

It is natural to ask what makes lower bound questions such as $P \stackrel{?}{=} PSPACE$ and $P \stackrel{?}{=} NC$ so difficult to solve. A non-technical reason for thinking they are difficult might be that some very bright people have tried and failed – but this is hardly satisfactory. A technical reason along the same lines would be provided by a reduction to these questions from another problem known to be really hard such as the Riemann Hypothesis. Perhaps the ultimate demonstration that $P \stackrel{?}{=} NP$ is a hard problem would be to show it to be independent of set theory (ZFC).

Another way to answer this question is to demonstrate that *known* methods are inherently too weak to solve problems such as $P \stackrel{?}{=} NP$. This approach was taken in Baker, Gill, and Solovay [4] who used oracle separation results for many major complexity classes to argue that relativizing proof techniques could not solve these problems. Since relativizing proof techniques involving diagonalization and simulation were the only available tools at the time of their work progress along known lines was ruled out.

Instead, people started to look at these problems in terms of non-uniform (= Boolean) complexity. Along these lines, many (non-relativizing) proof techniques have been discovered and used to prove lower bounds (see e.g. [7, 1, 27, 10, 31, 32, 28, 2, 25, 29, 33, 24, 5, 30, 18, 19, 11, 9, 13, 20, 3]). These techniques are highly combinatorial; they exist in a much larger variety than their recursion-theoretic predecessors.

In this paper we introduce the notion of a *natural* proof. We argue that *all lower bound proofs for non-monotone models known to us in non-uniform Boolean complexity either are natural or can be represented as natural.* We show that if a cryptographic hardness assumption is true, then *no natural proof can prove superpolynomial lower bounds for general circuits*, and show *unconditionally* that *no natural proof can prove exponential lower bounds on the circuit size of the discrete logarithm problem.*

Natural proofs form a natural hierarchy depending on the degree to which the combinatorial property involved in the proof is constructive. We show without using any cryptographic assumption that AC^0-natural proofs which are sufficient to prove the parity lower bounds of [7, 27, 10] are inherently incapable of proving the bounds for $AC^0[q]$-circuits of [33, 24, 5]. We also give a technical argument suggesting one reason that natural proofs are indeed natural: we show that every formal complexity measure which can prove superpolynomial lower bounds for a *single* function, can do so for *almost all functions.* This is one of the key requirements for a natural proof in our sense.

One application of natural proofs has been recently given in [23]. It was shown that in certain fragments of Bounded arithmetic any proof of superpolynomial lower bounds for

general circuits would naturalize, i.e., could be recast as a natural proof. Combined with the material contained in Section 4 of this paper, this leads to the independence of such lower bounds from these theories (assuming our cryptographic hardness assumption).

1.1. Notation and definitions

We denote by F_n the set of all Boolean functions in n variables. Most of the time, it will be convenient to think of $f_n \in F_n$ just as of a binary string of length 2^n called the *truth-table* of f_n.

The notation AC^k, NC^k is used in the standard sense for denoting non-uniform classes. $AC^0[m]$, TC^0 and $P/poly$ are the classes of functions computable by bounded-depth circuits allowing $MOD-m$ gates, bounded-depth circuits allowing threshold gates and unbounded-depth circuits over an unrestricted basis, respectively.

2. Natural proofs

We start by defining what we mean by a "natural combinatorial property"; natural proofs will be those that use a natural combinatorial property.

Formally, by a combinatorial property of Boolean functions we will mean a set of Boolean functions $\{C_n \subseteq F_n \mid n \in \omega\}$. Thus, some Boolean functions will possess property C_n and some will not. The combinatorial property C_n is *natural* if it contains a subset C_n^* with the following two conditions:

Constructivity: The predicate $f_n \stackrel{?}{\in} C_n^*$ is in P. Thus, C_n^* is computable in time which is polynomial in the truth table of f_n,

Largeness: $|C_n^*| \geq 2^{-O(n)} \cdot |F_n|$.

A combinatorial property C_n is *useful against* $P/poly$ if it satisfies:

Usefulness: The circuit size of any sequence of functions $f_1, f_2, \ldots, f_n, \ldots$, where $f_n \in C_n$, is superpolynomial, i.e., for any constant k, for sufficiently large n, the circuit size of f_n is greater than n^k.

A proof that some function does not have polynomial-sized circuits is *natural against* $P/poly$ if the proof contains, more or less explicitly, the definition of a natural combinatorial property C_n which is useful against $P/poly$.

Note that the notion of a natural proof, unlike that of a natural combinatorial property, is not quite precise. This is because while the notion of a property being explicitly defined in a journal paper is perfectly clear to the working mathematician, it is a bit slippery to formalize. When we make general statements about natural proofs (see Section 4), it will appear only in the context "there exists a natural proof..." and should be understood as equivalent to "there exists a natural combinatorial property C_n..."

The definitions of natural property and natural proof can be explained much less formally. A proof that some explicit[1] function $\{g_n\}$ does not have polynomial-sized circuits must work by stating some combinatorial property of Boolean functions, C_n, and proving that for any $f_n \in C_n$, the circuit size of f_n is superpolynomial in n^2. In other words, C_n is a combinatorial property of Boolean functions which implies that any function with that property is hard to compute; C_n is useful in the sense defined above. The proof would then continue by proving that g_n has property C_n; hence $\{g_n\} \notin P/poly$. If the function $\{g_n\}$ was provably in NP, then the proof could also conclude that $P \neq NP$.

For this proof to be natural against $P/poly$ the property C_n used by the proof must be itself natural, i.e., C_n must contain a subset C_n^* satisfying the constructivity and largeness conditions above (it will often turn out that $C_n^* = C_n$). The largeness condition says that C_n^* must be true of at least a polynomial (in 2^n) fraction of the entire universe F_n of Boolean functions in n variables. The constructivity condition requires that $f_n \in C_n^*$ can be decided by a Turing machine with running time polynomial in the truth table for the function f_n, i.e, polynomial-time in 2^n.

As it turns out, all combinatorial lower bounds for restricted non-monotone models work in exactly this way. In monotone models, the lower bounds use constructive combinatorial properties, but there is apparently no formal analogue of the largeness condition[3]. In Section 5, we give formal evidence that the largeness condition would apply to any useful C_n from a large class of combinatorial properties by showing that any formal complexity measure must have it. We have no formal evidence for constructivity, but a plausibility argument would be that we don't understand the mathematics of highly non-constructive C_n well enough to use them effectively in a proof. As will become clear in the examples, computing predicates in time polynomial in the length of *truth tables* of Boolean functions is not a strong restriction. It is also worth observing that for any C_n satisfying the largeness property, to be unnatural means not to contain any large, constructive subset. This is very similar to the notion of immune set.

[1] The reader can think of an explicit function as a function in NP.

[2] Any lower bound proof can be seen in this way, at least in the trivial sense that $C_n = \{g_n\}$.

[3] In particular, a useful definition of random monotone function is not evident.

The best example of a (supposedly) unnatural argument is a traditional counting argument. The combinatorial property C_n would just be something asserting that $\{f_n\}$ is not in $P/poly$ (e.g., $C_n(f_n) = 1$ exactly when the complexity of f_n is greater than $n^{\log n}$). The proof that C_n is large does not give us a least hint as to how to construct a large *constructive* subset $C_n^* \subseteq C_n$. Moreover, a consequence of Theorem 4.1 is that if a hardness assumption is true then such C_n^* can not exist at all! Thus, a counting argument is presumably not a natural argument. This poses no problem for us since counting arguments (closely associated with diagonalization arguments) have yet not proved any lower bounds for explicit functions.

All the lower bounds we study in this paper plainly state a natural property in their proofs. Thus, all these examples are natural proofs by our definition. In some cases, the fact that the stated property is natural is also evident from the original proof. In the less straightforward cases, the proof needs to be modified so that the new proof makes clear that the original proof was in fact natural. We will refer to modifying a proof so as to make clear that its associated property is natural as *naturalizing* the proof. This sometimes does require work (see e.g. Section 3.2.1 below). Given C_n, one must exhibit C_n^* and prove that it has both the constructivity and largeness conditions. The key to doing this seems to lie in carefully analyzing the lower bound proof that used C_n. In the case where a researcher intends to build a lower bound proof around some property C_n, evaluating C_n for naturalness might be non-trivial. In light of our framework, such an evaluation could be very useful.

For an insightful classification of the lower bound arguments obtained so far for restricted models we need to generalize the notions of natural property and natural proof.

Let Γ and Λ be complexity classes. Call a combinatorial property C_n Γ-*natural* if it contains $C_n^* \subseteq C_n$ with the following two conditions:

Constructivity: The predicate $f_n \stackrel{?}{\in} C_n^*$ is computable in Γ (recall, C_n^* is a set of truth-tables with 2^n bits),

Largeness: $|C_n^*| \geq 2^{-O(n)} \cdot |F_n|$.

A combinatorial property C_n is *useful against* Λ if it satisfies:

Usefulness: For any sequence of functions f_n, where the event $f_n \in C_n$ happens infinitely often, $\{f_n\} \notin \Lambda$.[4]

[4]Note that for the case $\Lambda = P/poly$ this general definition agrees with its more constructive version given above. A similar remark applies to all classes considered in this paper.

A lower bound proof that some explicit function is not in Λ is called Γ-*natural against* Λ if it states a Γ-natural property C_n which is useful against Λ.

Of interest to us will be proofs that are AC^0-natural, TC^0-natural, NC^k-natural, P-natural, and $P/poly$-natural. P-natural proofs will simply be called natural.

3. Examples of naturalizing arguments

3.1. AC^0 lower bounds for parity: AC^0-natural

One of the first combinatorial arguments to give people hope and direction in lower bound research was [7] where it was shown that $PARITY \notin AC^0$ (independently this result, using somewhat different machinery, was discovered in [1]). Substantial technical improvements to their bounds were subsequently given by [27, 10]. All these proofs are AC^0-natural.

The C_n used by these arguments simply says that there does not exist a restriction of the variables with the appropriate number of unassigned variables which forces f_n to a constant function. The "appropriate" number of unassigned variables is different in [7, 27, 10] and determines the bounds obtained.

All three papers argue explicitly that $C_n(f_n) = 1$ implies that $\{f_n\} \notin AC^0$, in other words, that C_n is useful against AC^0. C_n is a natural property. In fact, we can choose $C_n^* = C_n$.

A simple counting argument shows that C_n^* is true of a random function (C_n^* has the largeness condition).

C_n^* is in AC^0! (C_n^* has constructivity). Indeed, suppose k is the "appropriate" number of unassigned variables. Given the truth table for f_n as input, we compute $C_n^*(f_n)$ as follows. List all $\binom{n}{k}2^{n-k} = 2^{O(n)}$ restrictions of $n-k$ variables. For each one there is a circuit of depth 2 and size $2^{O(n)}$ which outputs a 1 iff that restriction does not leave f_n a constant function. Output the AND of all these circuits. The resulting circuit has depth 3 and is polynomial-sized in 2^n.

3.2. $AC^0[q]$ lower bounds: NC^2-natural

In this subsection we look at the proofs from [33, 24, 5] of lower bounds on the size of $AC^0[q]$-circuits, q being a power of a prime. The naturalness of these proofs is especially transparent in the framework of [33]. Namely, we have a $GF[2]$-linear mapping M from F_n to a matrix space, and we simply take C_n^* to be the set of all $f_n \in F_n$ for which $\mathrm{rk}(M(f_n))$ is large. After reading the argument in Section 3.2.1 below, it will be an exercise to show

that $C_n^*(f_n) = 1$ for at least $1/2$ fraction of all $f_n \in F_n$. Since computing the rank is in NC^2, we see that the proof is NC^2-natural. Smolensky's proof [24] is analyzed below.

We will show in Section 4 that *there is no AC^0-natural proof against $AC^0[2]$*. Along with the previous subsection, this *gives the insight that* [33, 24, 5] *had to require arguments from a stronger class than those of* [7, 27, 10].

3.2.1. Smolensky's proof: a non-trivial example of naturalization

The argument given in Smolensky [24] is a perfect example of a natural circuit lower bound proof, but this is not immediately obvious. We will outline a special case of his argument: a proof that parity does not have small $AC^0[3]$ circuits.

First, we recall the notion of polynomial approximation of a Boolean function. Think of the Boolean value TRUE as corresponding to the field element -1 and the Boolean value FALSE as corresponding to the field element 1. Let f be a Boolean function and p be a polynomial over Z_3 where f and p have an identical set of variable names. Any assignment A to f can be viewed as an assignment to p; in the case $p(A)$ and $f(A)$ evaluate to corresponding values we consider them equal on this assignment. Otherwise, we consider them to differ. The better p *approximates* f, the fewer assignments on which they differ. Since we will only be interested in the values polynomials take on $\{-1, 1\}$ (Boolean) assignments, we will consider polynomials to be multi-linear by default (no variable gets raised to a power greater than one).

Proof outline: Smolensky's proof has two main pieces. (1) Any function computed by a "small" $AC^0[3]$ circuit can be "reasonably" approximated by a "low" degree polynomial over Z_3. (2) The parity function in n variables can't be "reasonably" approximated by a "low" degree polynomial over Z_3. The proof of (1) is not important here and is omitted. (2) is proved by contradiction. Suppose there were a "low" degree (degree d) polynomial p which agrees with the polynomial $x_1 x_2 x_3 \cdots x_n$ (the parity function) on all but a "small" number of Boolean assignments. Let W be the set of Boolean assignments on which they differ. Let $N = 2^n$. Let w be the size of the set W. We will assume that n is odd and use l_1 and l_2 to denote polynomials of degree less than $n/2$. Every multi-linear polynomial q can be written in the form $x_1 \cdots x_n l_1 + l_2$. This means that, ignoring the inputs in W, every Z_3-valued function on $\{-1, 1\}^n \setminus W$ (and there are 3^{N-w} of them) can be represented in the form $pl_1 + l_2$. This representation has degree $(n-1)/2 + d$ which by a counting argument can't represent as many as 3^{N-w} functions. Contradiction.

This proof might seem to be exploiting a very particular fact about how the parity function is expressed as a polynomial; it is not obvious how this same proof would apply to a large fraction of functions. However, the proof technique *is* by its nature applicable

to a large fraction of functions.

There is one choice of C_n clear from the proof: $C_n(f_n) = 1$ if f_n can't be reasonably approximated by a low degree polynomial over Z_3 (for the appropriate definitions of reasonable and low). Part (1) of Smolensky's argument proves that C_n is useful against $AC^0[3]$. Why is C_n natural? To see it we have to make a choice of C_n^*.

The simple choice is $C_n^* = C_n$. It is fairly obvious that C_n^* satisfies the largeness condition. But what about constructivity? It is not clear at all.

Thus we sink deeper into the proof and try to put

$$C_n^*(f_n) = 1 \text{ if every polynomial } q \text{ can be written in the form } \bar{f}_n l_1 + l_2, \tag{1}$$

where $\bar{f}_n$ is the unique multi-linear polynomial representing f_n. Then we have constructivity.

In order to see this, denote by L the vector space of all polynomials of degree less than $n/2$, and by T the complementary vector space of all (multi-linear) polynomials without monomials of degree less than $n/2$. The whole polynomial space is then represented as the direct sum $L \oplus T$ and also, since n is odd, we have $\dim(L) = \dim(T) = N/2$. Now, $C_n^*(f_n) = 1$ iff the linear mapping $\pi_{f_n} : L \longrightarrow T$ taking $l \in L$ to the projection of $\bar{f}_n l \in L \oplus T$ onto T is one-to-one (the reader can check his understanding at this point by verifying that the parity function has this property). Thus checking that $C_n^*(f_n) = 1$ amounts to checking that a matrix easily computable from f_n is non-singular which can be done in NC^2.

For so chosen C_n^* the largeness condition also looks plausible. But we have no easy proof of it.

We turn around this difficulty by trying to extend the definition of (1) as much as we can (so that we'll have more functions satisfying it) while preserving its spirit (so that constructivity will also be preserved) and keeping the lower bound provided by it. A short examination shows that the definition

$$C_n^*(f_n) = 1 \text{ iff } \dim(\bar{f}_n L + L) \geq N\,(1/2 + \epsilon) \tag{2}$$

which for $\epsilon = 1/2$ is the same as (1), is actually as good as (1) itself for arbitrary fixed $\epsilon > 0$. Indeed, (2) implies that at least $3^{N(1/2+\epsilon)-w}$ functions on $\{-1,1\}^n \setminus W$ can be represented by a degree $(n-1)/2 + d$ polynomial, and the same counting argument still works.

But if we define C_n^* as in (2) with $\epsilon = 1/4$, we also have largeness! This immediately follows from the fact that for every $f_n \in F_n$ either $C_n^*(f_n) = 1$ or $C_n^*(x_1 \oplus \cdots \oplus x_n \oplus f_n) = 1$ (cf. the proof of Theorem 5.2 a) below).

To show this fact, note that if $\dim\left(\bar{f}_n L + L\right) \geq 3N/4$ then $C^*_n(f_n) = 1$. Otherwise we have

$$\begin{aligned}
&\dim\left((x_1 \cdots x_n \bar{f}_n L + L)/L\right) = \\
&\dim\left((x_1 \cdots x_n L + \bar{f}_n L)/\bar{f}_n L\right) \geq \\
&\dim\left((x_1 \cdots x_n L + \bar{f}_n L + L)/(\bar{f}_n L + L)\right) = \\
&\dim\left((T + L)/(\bar{f}_n L + L)\right) \geq N/4
\end{aligned}$$

(the first equality here comes from the observation that $(\bar{f}_n)^2 = 1$ and thus multiplying by $\bar{f}_n$ defines an automorphism of $L \oplus T$). This gives us $C^*_n(x_1 \oplus \ldots \oplus x_n \oplus f_n) = 1$.

So, C_n is an NC^2-natural property.

Smolensky's proof is the most difficult example of naturalization we have encountered in our analysis. On the other hand, it perfectly illustrates the general empirical idea of "adjusting" C_n in both directions in order to come up with required C^*_n.

3.3. Perceptron lower bounds for parity: P-natural

In [3], it is shown that a small constant-depth circuit (over $\{\wedge, \vee, \neg\}$) which is allowed a single majority gate can't approximate the parity function. They did this by first showing tight lower bound on the degree of a perceptron required to approximate parity to within a given ϵ. Their argument is natural.

Some definitions from [3]. A real polynomial p *strongly represents* a Boolean function[5] f just in case $\mathrm{sgn}(p(x)) = f(x)$ for all input vectors x; such a polynomial is also called a *perceptron* to compute f. Let p *weakly represent* f just in case p is not the constant zero function on $\{-1, 1\}^n$, and $\mathrm{sgn}(p(x)) = f(x)$ for all x where $p(x)$ is nonzero. The *weak degree*, $d_w(f)$, is defined as the least k for which there exists a non-zero degree k polynomial which weakly represents f.

A natural C_n stated in the paper is that f_n can't be well approximated by the sign of a low degree polynomial. It is explicitly shown that any f_n with property C_n can't be approximated by a small, constant-depth circuit with one majority gate, i.e., C_n has usefulness. To see that C_n is natural one must exhibit a proper subset C^*_n.

Let $C^*_n(f_n) = 1$ if $d_w(f_n)$ is greater than the appropriate threshold. [3] explicitly showed that $C^*_n(f_n) = 1$ implies that a polynomial must have appropriately high degree to

[5] in this section we, similarly to 3.2.1, represent Boolean functions as mappings from $\{-1, 1\}^n$ to $\{-1, 1\}$, and fg stands for the point-wise product which is the same as $f \oplus g$ in the $\{0, 1\}$-notation

approximate f_n with its sign, i.e., $C_n^*(f_n) = 1$ implies that $C_n(f_n) = 1$. d_w is computable in polynomial-time using linear programming. This shows that C_n^* has constructivity. Since the linear programming seems essential it is doubtful that anything substantially more constructive than C_n^* could be found in the above argument, e.g., an NC-natural property for example.

To argue that C_n^* has the largeness property, we can show the following improvement of an $\Omega(n/\log n)$ lower bound from [3]:

Theorem 3.1. *For almost all $f_n \in F_n$, $d_w(f_n) \geq n/20$.*

Proof. We use the following well-known facts:

Proposition 3.2. *Let $a_1, \ldots, a_N \in \mathbb{R}$. Then there exist $a'_1, \ldots, a'_N \in \mathbb{Z}$ such that $|a'_i| \leq \exp(O(N \log N))$ $(1 \leq i \leq N)$, and for every $x_i \in \{-1, 1\}^N$,*

$$\operatorname{sgn}\left(\sum_{i=1}^{N} a_i x_i\right) = \operatorname{sgn}\left(\sum_{i=1}^{N} a'_i x_i\right).$$

Proposition 3.3. *Every integer polynomial $p(x_1, \ldots, x_n)$ of degree d which is not an identical zero on $\{-1, 1\}^n$, differs from zero on at least 2^{n-d} points from $\{-1, 1\}^n$.*

The proof of Proposition 3.2 can be found e.g. in [16]; Proposition 3.3 is folklore.

Now, if f_n is weakly represented by a polynomial p of degree at most $n/20$, we firstly apply Proposition 3.2 to the vector of coefficients of p. The length N of this vector is $\sum_{i=0}^{n/20} \binom{n}{i} \leq 2^{n(\mathbf{H}(1/20)+o(1))}$, where $\mathbf{H}(\epsilon)$ is the entropy function. We find that p can be replaced by a polynomial p' with integer coefficients whose bit size is at most $O(N^2 \log N) \leq 2^{n(2 \cdot \mathbf{H}(1/20)+o(1))}$.

f_n can be uniquely retrieved from the pair (p', f'_n), where f'_n is the list of values of f_n on zeros of p' (arranged, say, in the lexicographic order). From Proposition 3.3 we know that the bit size of f'_n is at most $2^n - 2^{19/20n}$, thus the bit size of the pair (p', f'_n) is at most $2^n - 2^{19/20n} + 2^{n(2 \cdot \mathbf{H}(1/20)+o(1))}$. Since $2 \cdot \mathbf{H}(1/20) < \frac{19}{20}$, the proof is completed by the standard counting argument.■

3.4. Lower bounds on formula size: AC^0-natural

Andreev [30] gives a promising lower bound for the formula size of an explicit function. His bound was subsequently improved in [18, 19]. Finally, Hastad [11] gave a nearly optimal lower bound (almost n^3) of the formula size for Andreev's function.

Andreev's function is a Boolean function A_{2n} on $2n$ bits: $a_1, a_2, \ldots, a_n$; $b_1, b_2, \ldots, b_n$. The a's are partitioned into $\log n$ groups of size $n/\log n$ each. Let h_j be the parity of the bits in the jth group. The bits $h_1, h_2, \ldots, h_{\log n}$ index a number i from 1 to n. The value of the function A_{2n} is the bit b_i.

All these proofs work by using a shrinkage factor T which is successively improved in the last three papers until $T = \tilde{\Omega}(n^2)$. ($\tilde{\Omega}$ is the "soft Omega" notation which is like Ω but ignores multiplicative factors of $(\log n)^k$ for constant k.)

The meaning of T is that when a formula is hit by a random restriction it is almost certain to shrink by a factor of T. Thus, to prove a formula lower bound just show that a formula must have size s after being hit by a random restriction and it follows that the original formula had size around sn^2.

The natural property C_{2n} is that there is a restriction of b's such that any of its extensions leaving at least one unrestricted variable in each group of a's induces a formula of complexity $\Omega(n/\log n)$. This property is useful since a random restriction leaving $(\log n)^2$ unrestricted variables leaves at least one such variable in each group: for any fixing of b's, a random restriction to the a's will shrink the formula to $\Omega(n/\log n)$. Obviously, A_{2n} has C_{2n} (simply restrict b's so that they will encode the most complex function in $\log n$ variables) which implies that it must have formula complexity at least $\tilde{\Omega}(n^3)$.

We can choose $C^*_{2n} = C_{2n}$. The fact that C^*_{2n} has largeness is easy to prove. Constructivity is also easy if we observe that there are only $2^{O(n)}$ formulas of size less than $n/\log n$.

3.5. Lower bound against 2-levels of threshold functions: TC^0-natural

Hajnal et al. [9] show that the MOD-2 inner-product function requires depth-2 threshold circuits of exponential size. Any Boolean function can be viewed as a Boolean matrix by dividing the inputs into two equal sets with the left half indexing the rows and the right half indexing the columns. Seen in this way the inner-product function is a Hadamard matrix. Their proof shows that any matrix with low discrepancy can't be computed by small depth-2 threshold circuits. Choose C_n to be true of all functions whose matrices have low discrepancy. Their main lemma shows that any Hadamard matrix has low discrepancy.

The same argument shows that any matrix which is almost Hadamard in the sense that the dot product of any two rows or any two columns is small also has the low discrepancy property. Thus, the C_n^* suggested by their proof is to check that the function viewed as a matrix is almost Hadamard, for the appropriate definition of almost. It is possible to define "almost" so as to guarantee that C_n^* has largeness and preserves usefulness. Constructivity: For each of the $2^{O(n)}$ dot products, feed the pairs of AND's into a threshold gate; feed the outputs of the threshold gates into a large fan-in AND. This is in TC^0.

3.6. Lower bounds against switching-and-rectifier networks: AC^0-natural

It was shown in [34] that any switching-and-rectifier network (in particular, any nondeterministic branching program) for a large variety of symmetric functions must have size $\Omega(n\alpha(n))$, where $\alpha(n)$ is a function which slowly grows to infinity. A similar result was proven in [14] for $\oplus$-branching programs.

The proofs are based upon a purely combinatorial characterization of the network size in terms of particular instances of the MINIMUM COVER problem. Let C_n be the set of those functions f_n for which the size $\tau(f_n)$ of the minimal solution to the corresponding instance is $\Omega(n\alpha(n))$.

The key lemma in these proofs says that if f_n outputs a 1 on any input with $s(n)$ ones, and outputs a 0 on any input with $s(n) - d(n)$ ones, then $\tau(f_n) \geq \Omega(n\alpha(n))$ ($s(n)$ and $d(n)$ are functions which slowly grow to infinity.)

Denote this property by A_n. It obviously violates the largeness condition. We circumvent this by letting C_n^* be the set of those functions for which any restriction ρ assigning $n/2$ variables to zero can be extended to another restriction ρ' by assigning to zero $(n/2 - \log\log n)$ additional variables in such a way that the induced function has $A_{\log\log n}$.

To see $C_n^* \subseteq C_n$, recall from [34, 14] that every covering set $\delta_{i,\epsilon}(A)$ has its associated variable x_i such that restricting this variable to 0 kills $\delta_{i,\epsilon}(A)$. Now, for any collection of $o(n\alpha(n))$ covering sets we simply assign $n/2$ most frequently represented x_i's to 0, and this leaves us with a collection in which *every* variable corresponds to at most $o(\alpha(n))$ sets. Hence, for every extension ρ' of this restriction, the size of the resulting collection will be $o(\log\log n \cdot \alpha(n))$. Thus, by the above lemma, this collection (and hence the original one) does not cover all the points from the universe ($\alpha(n)$ and $\alpha(\log\log n)$ differ by at most 1).

C_n^* is in AC^0 (cf. Section 3.1).

To see the largeness condition, note that for every ρ we can choose $n^{3/2}$ extensions $\rho'_1, \ldots, \rho'_{n^{3/2}}$ so that the sets of variables unassigned by every two different ρ'_i, ρ'_j from this

list have at most one variable in common. Hence, the events "$\boldsymbol{f}_n$ restricted by ρ'_i has $A_{\log\log n}$" are independent (here $\boldsymbol{f}_n$ is a random function from F_n), and we can apply the standard counting argument.

4. Inherent limitations of natural proofs

In this section, we argue that natural proofs for lower bounds are *almost self-defeating*. The idea is that a natural proof that some function f is not in $P/poly$ has an associated algorithm. But just as the proof must distinguish f from a pseudo-random function in $P/poly$ (one being hard the other not), the associated algorithm must be able to tell the difference between the two. Thus, the algorithm can be used to break a pseudo-random generator. This is self-defeating in the sense that a natural proof that hardness exists would have as an automatic by-product an algorithm to solve a "hard" problem.

For a pseudo-random generator $G_n : \{0,1\}^n \longrightarrow \{0,1\}^{2n}$ define its *hardness* $H(G_n)$ as the minimal S for which there exists a circuit C of size $\leq S$ such that

$$|\mathbf{P}[C(G_n(\boldsymbol{x})) = 1] - \mathbf{P}[C(\boldsymbol{y}) = 1]| \geq \frac{1}{S}$$

(cf. [6]). Here, as usual, $\boldsymbol{x}$ is taken at random from $\{0,1\}^n$, and $\boldsymbol{y}$ is taken at random from $\{0,1\}^{2n}$.

Theorem 4.1. [6] *Assume that there exists a lower bound proof which is $P/poly$-natural against $P/poly$. Then for every polynomial time computable $G_k : \{0,1\}^k \longrightarrow \{0,1\}^{2k}$, $H(G_k) \leq 2^{k^{o(1)}}$.*

Equivalently, if 2^{n^ϵ}-hard functions exist then there is no $P/poly$-natural proof (against $P/poly$).

Proof. Let C_n be the $P/poly$-natural combinatorial property associated with the proof, and $C_n^* \subseteq C_n$ satisfy the constructivity and largeness conditions. W.l.o.g. we may assume from the very beginning that $C_n^* = C_n$.

We use a slightly modified construction from [8]. Let $G_k : \{0,1\}^k \longrightarrow \{0,1\}^{2k}$ be a polynomial time computable pseudo-random generator, and $\epsilon > 0$ be an arbitrary constant. Set $n \rightleftharpoons \lceil k^\epsilon \rceil$. We use $G : \{0,1\}^k \longrightarrow \{0,1\}^{2k}$ for constructing a pseudo-random function

[6] Razborov [23] has observed that a straightforward extension gives the same result for proofs using properties possessing largeness and computable by non-uniform circuits of size $2^{n^{O(1)}}$ (that is, quasipolynomial in 2^n)

generator $f : \{0,1\}^k \longrightarrow F_n$ in the same way as in [8]. Namely, let $G_0, G_1 : \{0,1\}^k \longrightarrow \{0,1\}^k$ be the first and the last k bits of G, respectively. For a string $y \in \{0,1\}^n$ we define $G_y : \{0,1\}^k \longrightarrow \{0,1\}^k$ by $G_y \rightleftharpoons G_{y_n} \circ G_{y_{n-1}} \circ \cdots \circ G_{y_1}$, and for $x \in \{0,1\}^k$ let $f(x)(y)$ be the first bit of $G_y(x)$.

Note that $f(x)(y)$ is polynomially time computable, hence for any fixed $x \in \{0,1\}^k$, the function $f(x) \in F_n$ is computable by polynomial size circuits. Hence, from the definition of a proof natural against $P/poly$, $f(x) \notin C_n$ (for sufficiently large k). This implies that C_n is a statistical test for $f(\boldsymbol{x})$ computable by circuits of size $2^{O(n)}$ and such that

$$|\mathbf{P}[C_n(\boldsymbol{f}) = 1] - \mathbf{P}[C_n(f(\boldsymbol{x})) = 1]| \geq 2^{-O(n)}. \tag{3}$$

Here $\boldsymbol{f}$ is a random function from F_n.

Constructing from this a statistical test for strings in our case is even simpler than in [8]. Namely, we arrange all internal nodes of the binary tree T of height n:

$$v_1, v_2, \ldots, v_{(2^n-1)}$$

in such a way that if v_i is a son of v_j then $i < j$. Let T_i be the union of subtrees of T made by $\{v_1, \ldots, v_i\}$ along with all leaves. For a leaf y of T let $v_i(y)$ be the root of the subtree in T_i containing y. Let $G_{i,y} \rightleftharpoons G_{y_n} \circ \cdots \circ G_{y_{n-h(i,y)+1}}$, where $h(i,y)$ is the distance between $v_i(y)$ and y. Finally, define the random collection $\boldsymbol{f}_i$ by letting $\boldsymbol{f}_i(y)$ be the first bit of $G_{i,y}\left(\boldsymbol{x}_{v_i(y)}\right)$, where $\boldsymbol{x}_v$ are taken from $\{0,1\}^k$ uniformly and independently for all roots v of trees from T_i.

Since $\boldsymbol{f}_0$ is $\boldsymbol{f}$, and $\boldsymbol{f}_{2^n-1}$ is $f(\boldsymbol{x})$, we have from (3) that for some i,

$$|\mathbf{P}[C_n(\boldsymbol{f}_i) = 1] - \mathbf{P}[C_n(\boldsymbol{f}_{i+1}) = 1]| \geq 2^{-O(n)}.$$

Fixing all $\boldsymbol{x}_v$ but $\boldsymbol{x}_{v_{i+1}}$ while preserving the bias, we see that $H(G_k) \leq 2^{O(n)} \leq 2^{O(k^\epsilon)}$. As ϵ was arbitrary, the result follows.■

The assumption that 2^{n^ϵ}-hard functions exist is quite plausible. For example, despite many advances in computational number theory, multiplication seems to provide a basis for a family of such functions (known factoring algorithms are sufficiently exponential).

Based upon lower bounds for the parity function, Nisan [17] constructed a very strong generator secure against AC^0-attack. In fact, an easy analysis of his generator from the point of computability gives the following:

Theorem 4.2. *For any integer d, there exists a family $G_{n,s} \subseteq F_n$, where s is a seed of size polynomial in n such that $G_{n,s} \in AC^0[2]$ and $G_{n,s}$ looks random for $2^{O(n)}$-size depth-d circuits, i.e., for any polynomial-size (in 2^n) depth d circuit family $C_n : F_n \longrightarrow \{0,1\}$,*

$$|\mathbf{P}[C_n(\boldsymbol{f}) = 1] - \mathbf{P}[C_n(G_{n,\boldsymbol{s}}) = 1]| < 2^{-\omega(n)}. \tag{4}$$

Here $\boldsymbol{f}$ is a random function from F_n and $\boldsymbol{s}$ is a random seed of the appropriate size.

Theorem 4.3. *There is no lower bound proof which is AC^0-natural against $AC^0[2]$.*

Proof. Assume, on the contrary, that such a proof exists, and let C_n has the same meaning as in the proof of Theorem 4.1. Let d be the depth of a size $2^{O(n)}$ circuit to compute C_n. Let $G_{n,s}$ be the generator which is pseudo-random against depth-d $2^{O(n)}$-sized circuits from Theorem 4.2. From the definition of a proof natural against $AC^0[2]$, for sufficiently large n, $C_n(G_{n,s}) = 0$. Now, (4) immediately contradicts the largeness condition.■

In fact, it is clear from the above proof that whenever a complexity class Λ contains pseudo-random function generators that are sufficiently secure against Γ-attack, then there is no Γ-natural proof against Λ.

4.1. Natural proofs are nor applicable to the discrete logarithm problem

It is possible (though we are unaware of any such examples) that a lower bound proof for restricted models might be natural, but can not be applied to any explicit function. In other words, the proof might simply argue that many functions are complex without providing us with any explicit examples of such functions. Given our hardness assumption, no natural proof can prove lower bounds against *P/poly* whether or not the proofs makes explicit what the hard function is. Avi Wigderson has pointed out that if we restrict ourselves to certain explicit functions, we can prove *unconditional* results in the style of Theorem 4.1. A good example of such a function is the discrete logarithm. The key point is that the discrete logarithm is known to be hard on average if and only if it is hard in the worst case. In this section, we prove that there is no natural proof that the discrete logarithm requires exponential-sized circuits.

Recall from [6] that for a prime p and a generator g for $\mathbb{Z}_p^*$, the predicate $B_{p,g}(x)$ on $\mathbb{Z}_p^*$ is defined to be 1 if $\log_g x \leq (p-1)/2$ and 0 otherwise. $B_{p,g}(x)$ was shown in [6] to be a hard bit of the discrete logarithm problem. We consider $B_{p,g}(x)$ as a Boolean function in $\lfloor \log p \rfloor$

variables (extended by, say, zeros on those inputs x which do not represent an integer in the range $[1, p-1]$). Our principal goal in this section is to show that *no P/poly-natural proof against "sufficiently large" Boolean circuits can be applied to* $B_{p,g}(x)$.

We need a couple of technical definitions. For an integer-valued function $t(n)$, let $SIZE(t(n))$ be the complexity class consisting of all functions $\{f_n\}$ which have circuit size $O(t(n))$. Let

$$t^{-1}(n) \rightleftharpoons \max\{x \mid t(x) \leq n\}.$$

We say that $t(n)$ is *half-exponential* if it is non-decreasing and

$$t^{-1}(n^C) \leq o(\log t(n)) \tag{5}$$

for every $C > 0$. The meaning of this definition is that, roughly speaking, the second iteration of $t(n)$ should grow faster than the exponent. For example, $t(n) = 2^{n^\epsilon}$ is half-exponential, whereas $t(n) = 2^{(\log n)^C}$ is not.

Theorem 4.4. *Let C_n be any P/poly-natural combinatorial property useful against $SIZE(t(n))$, where $t(n)$ is an arbitrary half-exponential function. Then $\bigcup_{n\in\omega} C_n$ may contain only finitely many functions of the form $B_{p,g}(x)$.*

Proof. Assume the contrary, and let $\{B_{p_\nu,g_\nu}\}$ be an infinite sequence contained in $\bigcup_{n\in\omega} C_n$ such that $\lfloor \log p_1 \rfloor < \lfloor \log p_2 \rfloor < \ldots$ Let $k_\nu \rightleftharpoons \lfloor \log p_\nu \rfloor$. Applying the usefulness condition to the sequence f_n obtained from $\{B_{p_\nu,g_\nu}\}$ by letting $f_n \equiv 0$ for those n which are not of the form $\lfloor \log p_\nu \rfloor$, we will find in $\{B_{p_\nu,g_\nu}\}$ an infinite subsequence where all functions have the circuit size at least $t(k_\nu)$. W.l.o.g. we may assume that this is the case for our original sequence.

Let $G_\nu : \{0,1\}^{2k_\nu} \longrightarrow \{0,1\}^{4k_\nu}$ be the standard pseudo-random generator from [6] based upon $\{B_{p_\nu,g_\nu}\}$. It is easy to check that the proof of [6, Theorem 3] actually extends to showing that the circuit size of $\{B_{p_\nu,g_\nu}\}$ is polynomial in $H(G_\nu) + k_\nu$. Thus, we have

$$t(k_\nu) \leq (H(G_\nu) + k_\nu)^{O(1)}. \tag{6}$$

Now we convert G_ν into the pseudo-random function generator $f_\nu : \{0,1\}^{2k_\nu} \longrightarrow F_{n_\nu}$ as in the proof of Theorem 4.1, where n_ν will be specified a little bit later. $f_\nu(x)(y)$ is computable by circuits of size $(k_\nu + n_\nu)^C$ for some $C > 0$. Let $n_\nu \rightleftharpoons t^{-1}(k_\nu^{C+1}) + 1$.

(5) implies that $t(k_\nu) > k_\nu^{C+1}$ for almost all ν, since otherwise we would have $k_\nu \leq t^{-1}\left(k_\nu^{C+1}\right) \leq \log t(k_\nu) \leq (C+1)\log k_\nu$. Hence $n_\nu \leq k_\nu$. Now we have that for almost all ν every function in the image of the generator f_ν has circuit size at most $(k_\nu + n_\nu)^C \leq$

$(2k_\nu)^C \leq k_\nu^{C+1} \leq t(n_\nu)$. Applying the usefulness condition again, we find that for almost all ν, the image of the generator f_ν has the empty intersection with C_n. Arguing as in the proof of Theorem 4.1, we get from this

$$H(G_\nu) \leq 2^{O(n_\nu)}. \tag{7}$$

Finally note that $C_n \neq \emptyset$ for almost all n (from largeness) and, thus,

$$t(n) \leq 2^n \tag{8}$$

(again, for almost all n.)

The required contradiction is now obtained simply by combining the inequalities (5) (with $n := k_\nu$, $C := C + 1$), (6), (7), (8):

$$n_\nu = t^{-1}(k_\nu^{C+1}) + 1 \leq o(\log t(k_\nu)) \leq o\left(\log H(G_\nu) + \log k_\nu\right) \leq o(n_\nu) + o(\log k_\nu) \leq o(n_\nu).$$

■

Corollary 4.5. *Let C_n be any P/poly-natural combinatorial property useful against $\bigcap_{\epsilon>0} DTIME\left(2^{n^\epsilon}\right)$. Then $\bigcup_{n\in\omega} C_n$ may contain only finitely many functions of the form $B_{p,g}(x)$.*

Proof. $\bigcap_{\epsilon>0} DTIME\left(2^{n^\epsilon}\right) \supseteq SIZE\left(2^{2^{\sqrt{\log n}}}\right)$, and $t(n) = 2^{2^{\sqrt{\log n}}}$ is half-exponential.■

It is easy to see that the above proofs are actually valid for an *arbitrary* collection $\{f_{p,g}\}$ of functions poly-time nonuniformly Turing reducible to the corresponding discrete logarithm problem in place of $\{B_{p,g}\}$.

5. One property of formal complexity measures

A *formal complexity measure* (see e.g. [26, Section 8.8], [21]) is an integer-valued function μ on F_n such that $\mu(f) \leq 1$ for $f \in \{\neg x_1, \ldots, \neg x_n, x_1, \ldots, x_n\}$ and $\mu(f * g) \leq \mu(f) + \mu(g)$ for all $f, g \in F_n$ and $* \in \{\wedge, \vee\}$. The meaning of this definition is that for every formal complexity measure μ, $\mu(f)$ provides a lower bound on the formula size of f, and actually many known lower bounds, both for monotone and non-monotone formulae, can be viewed from this perspective. See the above-cited sources for examples. Also, for any

approximation model $\mathfrak{M}$ (see [22] for the most general definition), we have $\rho(f * g, \mathfrak{M}) \leq \rho(f, \mathfrak{M}) + \rho(g, \mathfrak{M}) + 1$, hence $\rho(f, \mathfrak{M}) + 1$ is a formal complexity measure.

In this section we show that any formal complexity measure μ which takes a large value at a single function, must take large values almost everywhere. In particular, every combinatorial property based on such a measure automatically satisfies the largeness condition in the definition of natural property.

More specifically, we have the following:

Theorem 5.1. *Let μ be a formal complexity measure on F_n, and $\mu(f) = t$ for some $f \in F_n$. Then*:

a) *for at least $1/4$ fraction of all functions $g \in F_n$, $\mu(g) \geq t/4$;*

b) *for any $\epsilon = \epsilon(n)$ we have that for at least $(1 - \epsilon)$ fraction of $g \in F_n$,*

$$\mu(g) \geq \Omega\left(\frac{t}{\left(n + \log \frac{1}{\epsilon}\right)^2}\right) - n.$$

In fact, the main argument used in the proof of this theorem is valid for arbitrary Boolean algebras, and we formulate it as a separate result since this might be of independent interest.

Theorem 5.2. *Let B be a finite Boolean algebra with N atoms and $S \subseteq B$.*

a) *if $|S| > \frac{3}{4}|B|$ then every element of B can be represented in the form*

$$(s_1 \wedge s_2) \vee (s_3 \wedge s_4); \ s_i \in S \ (1 \leq i \leq 4); \tag{9}$$

b) *assume additionally that S contains all atoms and coatoms of B. Then every element of B can be represented in the form*

$$\bigvee_{i=1}^{\ell} \bigwedge_{j=1}^{\ell} s_{ij}, \tag{10}$$

where $s_{ij} \in S$ and $\ell \leq O\left(\log \frac{N \cdot |B|}{|S|}\right)$.

Proof of Theorem 5.1 from Theorem 5.2. Let $S \rightleftharpoons \{g \mid \mu(g) < t/4\}$ for part a), and $S \rightleftharpoons \left\{ g \,\middle|\, \mu(g) \le \delta \cdot \frac{t}{\left(n+\log \frac{1}{\epsilon}\right)^2} \right\}$, where δ is a sufficiently small constant, for part b). Note that in part b) we may assume that $\delta \cdot \frac{t}{\left(n+\log \frac{1}{\epsilon}\right)^2} \ge n+1$ since otherwise there is nothing to prove. Since $\mu\left(\bigwedge_{i=1}^{n} p_i\right) \le n$ and $\mu\left(\bigvee_{i=1}^{n} p_i\right) \le n$, where p_i is either x_i or $\neg x_i$, this implies that S contains all atoms and coatoms of F_n, the latter being viewed as a Boolean algebra.

Now, if $|S| > \frac{3}{4}|B|$ in part a) or $|S| \ge \epsilon|B|$ in part b), then we would apply Theorem 5.2 and represent f in the form (9), (10) respectively. This representation in both cases would imply the bound $\mu(f) < t$, the contradiction.■

Now we prove Theorem 5.2. Denote by $\boldsymbol{b}$ a randomly chosen element of B.

Proof of Theorem 5.2 a). Fix $b_0 \in B$ and consider the representation

$$b_0 = (\boldsymbol{b} \wedge (\neg \boldsymbol{b} \oplus b_0)) \vee (\neg \boldsymbol{b} \wedge (\boldsymbol{b} \oplus b_0)) .$$

As all four random variables $\boldsymbol{b}, (\neg \boldsymbol{b} \oplus b_0), \neg \boldsymbol{b}, (\boldsymbol{b} \oplus b_0)$ are uniformly distributed on B and $|S| > \frac{3}{4}|B|$, for at least one particular choice b of $\boldsymbol{b}$ we have $b, (\neg b \oplus b_0), \neg b, (b \oplus b_0) \in S$.■

For proving part b) of Theorem 5.2 we need the following

Lemma 5.3. *Let B be a finite Boolean algebra with N atoms and $S \subseteq B$. Then there exists a subset $S_0 \subseteq S$ of cardinality $O(\log N)$ such that $\wedge S_0$ contains at most $O\left(\log \frac{|B|}{|S|}\right)$ atoms.*

Proof of Lemma 5.3. Let us call an atom a *good* if $\mathbf{P}[a \le \boldsymbol{s}] \le 2/3$ and *bad* otherwise. Here $\boldsymbol{s}$ is picked at random from S.

Now, the standard entropy-counting argument gives us that there are at most

$$O\left(\log \frac{|B|}{|S|}\right)$$

bad atoms. An equally standard argument implies that if we take a random subset $\boldsymbol{S}_0 \subseteq S$ of cardinality $C \log N$, the constant C being sufficiently large, then for any good atom a, $\mathbf{P}[a \le \wedge \boldsymbol{S}_0] < N^{-1}$. Hence, for at least one particular choice S_0 of $\boldsymbol{S}_0$, $\wedge S_0$ contains only bad atoms, and the lemma follows.■

Proof of Theorem 5.2 b). Denote $\frac{|S|}{|B|}$ by ϵ. Once again, fix $b_0 \in B$. Let us call $c \le b_0$ *good* if $\mathbf{P}[\boldsymbol{b} \in S \mid \boldsymbol{b} \wedge b_0 = c] \ge \frac{\epsilon}{2}$ and *bad* otherwise. Note that $\boldsymbol{b} \wedge b_0$ is *uniformly*

distributed on the Boolean algebra $B_0 \rightleftharpoons \{\, c \mid c \leq b_0 \,\}$. Hence

$$\mathbf{P}[\boldsymbol{c} \text{ is good}] \geq \frac{\epsilon}{2}, \tag{11}$$

where $\boldsymbol{c}$ is chosen from B_0 at random.

Now, fix a good $c \in B_0$. The set $B(c) \rightleftharpoons \{\, b \in B \mid b \wedge b_0 = c \,\}$ is a Boolean algebra. Applying to this algebra and to $S := S \cap B(c)$ Lemma 5.3, we come up with $S_0 \subseteq S$ of cardinality $O(\log N)$ such that $c \leq \wedge S_0$ and $(\wedge S_0 \backslash c)$ has at most $O\left(\log \frac{1}{\epsilon}\right)$ atoms. We extend S_0 by including to it the corresponding coatoms and find that every good $c \in B_0$ can be represented in the form $\bigwedge_{j=1}^{\ell} s_j,\ s_j \in S,\ \ell \leq O\left(\log \frac{N}{\epsilon}\right)$.

Next we apply the dual version of Lemma 5.3 to the Boolean algebra B_0 and $S := \{\, c \in B_0 \mid c \text{ is good} \,\}$. In view of (11), the same argument as above yields that $b_0 = \bigvee_{i=1}^{\ell} c_i$, where c_i are either good or atoms. The statement follows.■

6. Conclusion

We do not conclude that researchers should give up on proving serious lower bounds. Quite the contrary, by classifying a large number of techniques that are unable to do the job we hope to focus research in a more fruitful direction. Pessimism will only be warranted if a long period of time passes without the discovery of a non-naturalizing lower bound proof.

As long as we use natural proofs we have to cope with a duality: *any lower bound proof must implicitly argue a proportionately strong upper bound.* In particular, we have shown that a natural proof against complexity class Λ implicitly shows that Λ does not contain strong pseudo-random function generators. In fact, the proof gives an algorithm to break any such generator. Seen this way, even a natural proof against NC^1 (or TC^0) becomes difficult or impossible. In [12] it is argued based on the hardness of subset sum that a pseudo-random function exists in $TC^0 \subseteq NC^1$. Consider the plausible conjecture that there exists a (pseudo-random) function $f \in NC^1$ (or TC^0) such that $G_{n,s}(x) = f(s\#x)$ is a pseudo-random function generator. A natural proof that $P \neq NC^1$ or $P \neq TC^0$ would give an algorithm to break it. Thus, we see that working on lower bounds using natural methods is like breaking a secret code determined by the class we are working against!

With this duality in mind, it is no coincidence that the technical lemmas of [10, 24, 33] yield much of the machinery for the learning result of [15].

7. Acknowledgements

We would like to thank Oded Goldreich, Russell Impagliazzo, Mauricio Karchmer, Silvio Micali, Robert Solovay, and Avi Wigderson for very helpful discussions.

References

[1] M. Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, May 1983.

[2] N. Alon and R. Boppana. The monotone circuit complexity of Boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[3] J. Aspnes, R. Beigel, M. Furst, and S. Rudich. The expressive power of voting polynomials. In *Proceedings of the 23rd ACM STOC*, pages 402–409, 1991. Journal version to appear in *Combinatorica*.

[4] T.P. Baker, J. Gill, and R. Solovay. Relativizations of the $P = NP$ question. *SIAM Journal on Computing*, 4:431–442, 1975.

[5] D. A. Barrington. A note on a theorem of Razborov. Technical report, University of Massachusetts, 1986.

[6] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM Journal on Computing*, 13:850–864, 1984.

[7] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits and the polynomial time hierarchy. *Math. Syst. Theory*, 17:13–27, 1984.

[8] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.

[9] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, G. Turan. Threshold circuits of bounded depth. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 99–110, 1987.

[10] J. Håstad. *Computational limitations on Small Depth Circuits.* PhD thesis, Massachusetts Institute of Technology, 1986.

[11] J. Håstad. The shrinkage exponent is 2. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 114–123, 1993.

[12] R. Impagliazzo and M. Naor. Efficient cryptographic schemes provably as secure as subset sum. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 236–243, 1989.

[13] M. Karchmer and A. Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the 20th Annual ACM Symposium on the Theory of Computing*, pages 539–550, 1988.

[14] M. Karchmer and A. Wigderson. On span programs. In *Proceedings of the 8th Structure in Complexity Theory Annual Conference*, pages 102–111, 1993.

[15] N. Linial, Y. Monsour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science*, pages 574–579, 1989.

[16] S. Muroga. *Threshold logic and its applications.* Wiley-Interscience, 1971.

[17] N. Nisan. Pseudorandom bits for constant depth circuits. In *Combinatorica* 11 (1), pp. 63–70, 1991.

[18] N. Nisan and R. Impagliazzo. The effect of random restrictions on formula size. *Random Structures and Algorithms*, Vol. 4, No. 2, 1993.

[19] M. S. Paterson and U. Zwick. Shrinkage of de Morgan formulae under restriction. In *Proceedings of the 32th IEEE Symposium on Foundations of Computer Science*, pages 324–333, 1991.

[20] R. Raz and A. Wigderson. Monotone circuits for matching require linear depth. In *Proceedings of the 22th Ann. ACM Symposium on the Theory of Computing*, pages 287–292, 1990.

[21] A. Razborov. On submodular complexity measures. In M. S. Paterson, editor, *Boolean Function Complexity. London Math. Soc., Lecture Note Series* 169, pages 76–83. Cambridge University Press, 1992.

[22] A. Razborov. On small size approximation models. To appear in the volume *The Mathematics of Paul Erdos*, 1993.

[23] A. Razborov. Unprovability of lower bounds on circuit size in certain fragments of Bounded Arithmetic. To appear in *Izvestiya of the RAN*, 1994.

[24] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 77–82, 1987.

[25] É. Tardos. The gap between monotone and nonmonotone circuit complexity is exponential. *Combinatorica*, 8:141–142, 1988.

[26] I. Wegener. *The complexity of Boolean functions.* Wiley-Teubner, 1987.

[27] A. Yao. Separating the polynomial-time hierarchy by oracles. In *Proceedings of the 26th IEEE FOCS*, pages 1–10, 1985.

[28] А.Е. Андреев. Об одном методе получения нижних оценок сложности индивидуальных монотонных функций. *ДАН СССР*, 282(5):1033–1037, 1985. A. E. Andreev, On a method for obtaining lower bounds for the complexity of individual monotone functions. *Soviet Math. Dokl.* 31(3):530-534, 1985.

[29] А.Е. Андреев. Об одном методе получения эффективных нижних оценок монотонной сложности. *Алгебра и логика*, 26(1):3–21, 1987. A.E. Andreev, On one method of obtaining effective lower bounds of monotone complexity. *Algebra i logika*, 26(1):3-21, 1987. In Russian.

[30] А.Е. Андреев. О методе получения более чем квадратичных нижних оценок для сложности π-схем. *Вестник МГУ, сер. матем и механ.*, 42(1):63–66, 1987. A.E. Andreev, On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Moscow Univ. Math. Bull.* 42(1):63-66, 1987. In Russian.

[31] А. А. Разборов. Нижние оценки монотонной сложности некоторых булевых функций. *ДАН СССР*, 281(4):798–801, 1985. A. A. Razborov, Lower bounds for the monotone complexity of some Boolean functions, *Soviet Math. Dokl.*, 31:354-357, 1985.

[32] А. А. Разборов. Нижние оценки монотонной сложности логического перманента. *Матем. Зам.*, 37(6):887–900, 1985. A. A. Razborov, Lower bounds of monotone complexity of the logical permanent function, *Mathem. Notes of the Academy of Sci. of the USSR*, 37:485-493, 1985.

[33] А. А. Разборов. Нижние оценки размера схем ограниченной глубины в полном базисе, содержащем функцию логического сложения. *Матем. Зам.*, 41(4):598–607, 1987. A. A. Razborov, Lower bounds on the size of bounded-depth networks over a complete basis with logical addition, *Mathem. Notes of the Academy of Sci. of the USSR*, 41(4):333-338, 1987.

[34] А. А. Разборов. Нижние оценки сложности реализации симметрических булевых функций контактно-вентильными схемами. *Матем. Зам.*, 48(6):79–91, 1990. A. A. Razborov, Lower bounds on the size of switching-and-rectifier networks for symmetric Boolean functions, *Mathem. Notes of the Academy of Sci. of the USSR.*