Journal of Mathematical Logic, Vol. 11, No. 1 (2011) 11–27
© World Scientific Publishing Company
DOI: 10.1142/S0219061311000979

ON THE PROOF COMPLEXITY OF THE NISAN–WIGDERSON GENERATOR BASED ON A HARD NP ∩ coNP FUNCTION

JAN KRAJÍČEK

Department of Algebra
Faculty of Mathematics and Physics
Charles University
Sokolovská 83, Prague 8, CZ – 186 75
The Czech Republic
krajicek@karlin.mff.cuni.cz

Received 15 April 2010

Let g be a map defined as the Nisan–Wigderson generator but based on an NP ∩ coNP-function f. Any string b outside the range of g determines a propositional tautology $\tau(g)_b$ expressing this fact. Razborov [27] has conjectured that if f is hard on average for P/poly then these tautologies have no polynomial size proofs in the Extended Frege system EF.

We consider a more general Statement (S) that the tautologies have no polynomial size proofs in any propositional proof system. This is equivalent to the statement that the complement of the range of g contains no infinite NP set.

We prove that Statement (S) is consistent with Cook' s theory PV and, in fact, with the true universal theory T_{PV} in the language of PV. If PV in this consistency statement could be extended to "a bit" stronger theory (properly included in Buss's theory S^1_2) then Razborov's conjecture would follow, and if T_{PV} could be added too then Statement (S) would follow.

We discuss this problem in some detail, pointing out a certain form of reflection principle for propositional logic, and we introduce a related feasible disjunction property of proof systems.

Keywords: Proof complexity; Nisan–Wigderson generator.

Mathematics Subject Classification 2010: 03F20, 68Q17, 03H15

1. Introduction

A *propositional proof system* is a polynomial time function P whose range is exactly the set of propositional tautologies TAUT (say in 3DNF). This definition of Cook and Reckhow [8] captures the usual logical calculi for propositional logic: map a string that is a valid proof to the formula it proves and all other strings to $p \vee \neg p$. The range of such a map is TAUT by the soundness and the completeness of the calculus, and it can be computed in polynomial time because in logical calculi valid proofs can be recognized in polynomial time.

A P-proof of a formula φ is any string π such that $P(\pi) = \varphi$. A proof system P is *p-bounded* if there is a constant $c \geq 1$ such that any tautology φ has a P-proof of size at most $|\varphi|^c$. Cook and Reckhow [8] have observed that a p-bounded proof system exists if and only if the class NP is closed under complementation. It is therefore believed that no such proof system exists and to establish this is a fundamental problem of proof complexity.

At present it is not ruled out that the usual textbook propositional calculus based on a finite number of axiom schemes and inference rules (a *Frege system F* in the terminology of [8]) is p-bounded. It is generally assumed that the so called *Extended Frege system EF* is a pivotal case in the study of the fundamental problem. EF augments F by the ability to abbreviate formulas or, equivalently, it is a Frege system but operating with circuits rather than with formulas (cf. [13, 12]). The field of proof complexity has also other sources motivating its investigations (e.g. bounded arithmetic or automated theorem proving) and the interested reader may consult [13, 25].

A key issue in attacking the fundamental problem of the existence of a p-bounded proof system is to come up with plausible candidates for hard formulas, tautologies that it will be hard to prove in a proof system under consideration (or in all proof systems). A class of such candidate formulas, the so called *τ-formulas* or *proof complexity generators*, were proposed independently in [2, 14]. Their theory was developed so far in about a dozen papers and its large part (but not all) is exposed in [18, Chaps. 29 and 31]. We shall explain now the main idea.

Let $g : \{0,1\}^n \to \{0,1\}^m$ with $m = m(n) > n$ be a map. For simplicity we assume that $m(n)$ is injective, hence m determines n. Let $b \in \{0,1\}^m$ be any string outside of the range $\mathrm{Rng}(g)$ of g. If g is polynomial time or at least (as in this paper) an $\mathrm{NP} \cap \mathrm{coNP}$ map then the statement $b \notin \mathrm{Rng}(g)$ is a *co*NP property of b and can be expressed by a propositional formula $\tau(g)_b$ in the sense that

$$\tau(g)_b \in \mathrm{TAUT} \quad \text{if and only if} \quad b \notin \mathrm{Rng}(g).$$

We shall not discuss how precisely is the formula defined and take it as a canonical construction in the style of a proof of the NP-completeness of SAT, cf. [16].

Say that g is *a hard proof complexity generator for P* if and only if the $\tau(g)$-formulas have no polynomial size proofs in P: for any constant $c \geq 1$, if m is large enough then $\tau(g)_b$ has a P-proof of size $\leq |\tau(g)_b|^c$ for no $b \in \{0,1\}^m \backslash \mathrm{Rng}(g)$. The idea underlying the theory of these formulas is that there are maps that are hard for very strong proof systems like EF or maybe even for all proof systems.

There are several maps that are candidates for hard proof complexity generators (see [18]) but in this paper we shall consider only one example proposed by Razborov [27]. He considers a map g defined as the Nisan–Wigderson generator with parameters as in [23]. Such a map is determined by a matrix A and by a Boolean function f (see Sec. 2 for details). In [23] f is computed by a deterministic algorithm and is assumed to have a certain hardness property. Razborov's idea was to take, for the purpose of proof complexity, an NP $\cap$ coNP function f that is hard for all P/poly:

Conjecture 1.1 (Razborov [27, Conjecture 2]). *Any NW-generator based on a matrix A which is a combinatorial design with the same parameters as in [23] and on any function f in NP$\cap$ coNP that is hard on average for P/poly, is hard for EF.*

We will discuss the details (e.g. the conditions on the parameters involved) in Sec. 2. Pich [24] has recently established this statement for proof systems admitting a certain form of feasible interpolation, including resolution.

This is a beautiful conjecture. But it is not clear (to this author) why the statement should hold only for EF and not for other, or all, proof systems. Polynomial size EF proofs operate with polynomial size circuits but an arbitrary proof system can by p-simulated by one that has the same property (an extension of EF by polynomial time set of extra axioms, see [19, 13]). In particular, consider the following.

Statement (S)(informally). *Let g be the Nisan–Wigderson generator with the same parameters as in Razborov's conjecture, and assume it is based on an NP $\cap$ coNP-function f that is a hard bit of a one way permutation.*

Assume that R is an infinite NP-set which has infinitely many elements with length equal to $m(k)$ for some $k \geq 1$.

Then

$$\mathrm{Rng}(g) \cap R \neq \emptyset.$$

In this paper, we prove that Statement (S) is consistent with Cook's theory PV and, in fact, with the true universal theory T_{PV} in the language of PV. These two theories are significant in proof complexity: PV corresponds to EF and T_{PV} corresponds to the union of all proof systems. The correspondence we refer to here is a variety of technical results linking theories and proof systems (cf. [13, 7]). In particular, one can view PV as a uniform version of EF in a manner analogous to how polynomial time algorithms are uniform versions of polynomial size circuits.

Demonstrating that a complexity-theoretic conjecture is consistent with a bounded arithmetic theory appears to be interesting (as opposed to demonstrating the unprovability in a weak theory). The reason is that such a consistency result in effect establishes the conjecture in a structure (a model of the theory in question) that is quite close to the world of complexity theory. Even weak theories of bounded arithmetic (low in the Buss's hierarchy of theories S^i_2 and T^i_2, cf. [4, 13]) contain a significant part of contemporary complexity theory.

In our case there is an extra reason to consider a consistency statement like the one we do. If PV in this consistency statement could be extended to "a bit" stronger theory (properly included in Buss's theory S^1_2), then Razborov's conjecture would follow, and if T_{PV} could be added too then Statement (S) would follow.

This paper stems from a forcing construction in [18, Chap. 31] establishing a special case of our main theorem. Here we employ classic methods and give a new construction, and also extend the result to a larger variation of parameters in Statement (S) in Theorem 4.2. We also discuss in some detail what is the obstacle to extending the result in a way that would allow to deduce the Razborov's conjecture and possibly also Statement (S).

2. An Interactive Computational Task

We first briefly review the Nisan–Wigderson construction from [23], fixing the notation along the way. Let $1 \leq d \leq \ell \leq n < m$ be some parameters and let A be an $m \times n$ 0-1 matrix with ℓ ones per row. $J_i(A) := \{j \in [n] | \ A_{ij} = 1\}$. Let $f : \{0,1\}^\ell \to \{0,1\}$ be a Boolean function.

Define function $NW_{A,f} : \{0,1\}^n \to \{0,1\}^m$ as follows: The ith bit of the output is computed by f from the bits of the input that belong to $J_i(A)$. For x a string of length n and $J \subseteq [n]$ of size ℓ denote by $x(J)$ the substring of x of length ℓ consisting of those x_j from x for which $j \in J$. Hence the ith output bit of $NW_{A,f}(x)$ is $f(x(J_i(A)))$.

Matrix A is a (d, ℓ)-design if in addition the intersection of any two different rows $J_i(A) \cap J_k(A)$ has size at most d. Nisan and Wigderson [23] construct matrices A that are (d, ℓ)-designs for a wide range of parameters. In particular, m can be exponential in $n^{\Omega(1)}$ and this is crucial for many applications. For our purposes however, it is best to have m as small as possible and we shall fix the parameters as follows:

$$m := n + 1, \quad d := \log(n + 1), \quad \text{and} \quad \ell := n^{1/3}. \tag{2.1}$$

Later we shall remark on how the parameter m can be altered.

We shall denote by A_n some canonical matrix with these parameters (e.g. provided by one of the constructions from [23]). It is not important for us to have A_n explicit.

Let f be an NP ∩ *co*NP function (i.e. it is the characteristic function of a language in NP ∩ *co*NP). We shall assume that f is given by two NP predicates

$$\exists\, y(|y| \leq |u|^{O(1)} \wedge F_0(u, y)) \quad \text{and} \quad \exists\, y(|y| \leq |u|^{O(1)} \wedge F_1(u, y)) \tag{2.2}$$

with F_0 and F_1 polynomial-time relations such that

$$f(u) = a \quad \text{if and only if } \exists\, y(|y| \leq |u|^{O(1)} \wedge F_a(u, y))$$

for $a = 0, 1$. Any string y witnessing the existential quantifier will be called *a witness for $f(u) = a$*. We shall say that f has *unique witnesses* if the witness y is unique for all u.

Let A be a matrix and f an NP ∩ *co*NP function as above. As $n < m$ there are strings in $\{0,1\}^m$ that are outside of the range of $NW_{A,f}$. Fix $b = (b_1, \ldots, b_m) \in \{0,1\}^m$ any such string. In this situation we define the following.

Computational Task (T). *Given $x \in \{0,1\}^n$ find $i \in [m]$ such that the ith bit of $NW_{A,f}(x)$ differs from b_i.*

We shall consider a specific model for solving (T) in which two players, a computationally limited Student and an unrestricted Teacher, interact in the following way. In the first step:

- The Student, upon receiving an input $x \in \{0,1\}^n$, computes his first candidate solution $i_1 \in [m]$.
- If i_1 solves (T) the Teacher will acknowledge it and the computation stops.
- If i_1 fails to solve (T) the Teacher sends to the Student a witness y_1 to $f(x(J_{i_1}(A)) = b_{i_1}$.

In general, in the kth step the Student computes a candidate solution $i_k \in [m]$ from x and from the witnesses $y_1, \ldots, y_{k-1}$ he has received from the Teacher in the previous $k-1$ steps. The Teacher acts as above: if i_k solves (T) she acknowledges it, if not she sends to the Student a witness y_k certifying the incorrectness, i.e. witnessing $f(x(J_{i_k}(A))) = b_{i_k}$.

This computational model was introduced in [22] as an interpretation of a form of Herbrand theorem, and formalized in terms of computational classes in [21] (see also [13]).

For $c \geq 1$ we say that a Student *solves* (*T*) *in* c *steps* if the computation with any (honest) Teacher stops in at most c steps on every input $x \in \{0,1\}^n$. It is convenient to think of such a Student as being determined by c functions

$$S_1(x), S_2(x, y_1), \ldots, S_c(x, y_1, \ldots, y_{c-1}), \tag{2.3}$$

S_k computing the kth candidate solution i_k from x and from the witnesses $y_1, \ldots, y_{k-1}$ received from the Teacher in earlier rounds. We shall concentrate on the case when all S_k are computed by circuits C_k and we will be interested in the total size of these c circuits.

3. The Hardness of Task (T)

The following hypothesis will play a crucial role later on.

Hardness Assumption (H). *There is an NP ∩ coNP function* f *such that for all* $c \geq 1$ *and* $k \geq 1$ *the following holds for all large enough* n *and any* $b \in \{0,1\}^m \setminus \mathrm{Rng}(NW_{A_n,f})$:

Any circuits computing the moves of a Student that solves (*T*) *in* c *steps must have the total size at least* n^k.

This is a conservative formulation. One may contemplate a hypothesis that each Student solving (T) in a constant number of steps must have an exponential size (see Corollary 3.3).

In the rest of this section we shall derive (H) from a more conventional hypothesis. Recall the hardness of a Boolean function f used in [23]: For two number parameters $\epsilon(\ell)$ and $S(\ell)$ depending on ℓ define f to be (ϵ, S)-hard if for every ℓ

and every circuit C with ℓ inputs and of size at most $S(\ell)$ it holds:

$$\mathsf{Prob}_{u\in\{0,1\}^\ell}[C(u) = f(u)] < 1/2 + \epsilon/2,$$

[23] then use the concept for $\epsilon := 1/S$ and are concerned with the maximal S such that the function is $(1/S, S)$-hard; such S is called the *hardness* of f and denoted $H_f(\ell)$. We shall not use this setting of ϵ (in the main case of [23] m is exponential in n and that leads to exponentially small ϵ but our m is small). Instead we are interested mainly in the parameter S, with ϵ being always of the rate $\ell^{-O(1)}$. The only exception is part 3 in Theorem 4.2.

It is convenient to introduce the following notation (AH stands for *approximating hardness*) measuring the rate of the parameter S. Function $AH_f : \mathbf{N} \times \mathbf{N} \to \mathbf{N}$ is defined in the following way:

- For $\ell, k \geq$, define $AH_f(\ell, k)$ to be the minimal s such that there is a size s circuit C with ℓ inputs such that

$$\mathsf{Prob}_{u\in\{0,1\}^\ell}[C(u) = f(u)] \geq 1/2 + \ell^{-k}.$$

The difference $\mathsf{Prob}_{u\in\{0,1\}^\ell}[C(u) = f(u)] - 1/2$ will be called *the advantage* (tacitly, over $1/2$) of C in computing f.

In the definition of the task (T), we have not assumed that f, an NP $\cap$ *co*NP function, has unique witnesses but we shall use such an assumption in Theorem 3.2. Such functions do appear quite naturally as hard bits of polynomial-time permutations. Let us recall briefly the relevant notions (see [3, 10] for details) and state a formal lemma for a later reference. We use the non-uniform setting and do not stress it further in the terminology.

A polynomial time function h is a permutation if it permutes each $\{0,1\}^\ell$. It is defined to be $\epsilon(\ell)$ *one way with security parameter* $t(\ell)$ if and only if for all ℓ and any circuit D with ℓ inputs and of size at most $t(\ell)$ it holds:

$$\mathsf{Prob}_{v\in\{0,1\}^\ell}[D(h(v)) = v] \leq \epsilon(\ell).$$

There are several permutations constructed from discrete logarithm, factoring or RSA that are conjectured to be ℓ^{-k} one way with super-polynomial (or even exponential) security parameter (see [3, 10]). Having such a permutation h one may assume (by the Goldreich–Levin theorem, cf. [3]) without loss of generality that h has a hard bit function $b(v)$. That is, a small circuit (of size $t(\ell)^{\Omega(1)}$) can compute $b(v)$ from the input $u := h(v)$ only with a negligible advantage over $1/2$. Then we can define an NP $\cap$ coNP function f with unique witnesses by $f(u) := b(h^{(-1)}(u))$. This yields the following lemma.

Lemma 3.1. *Assume that there exists a polynomial time permutation h such that for any fixed $k \geq 1$ it is ℓ^{-k} one way with a super-polynomial security parameter $t(\ell) = \ell^{\omega(1)}$.*

Then there exists an NP $\cap$ coNP function f with unique witnesses such that for any fixed $k \geq 1$ the function $AH_f(\ell, k)$ is a super-polynomial function of ℓ.

If, in fact, for any fixed $k \geq 1$ h is even ℓ^{-k} one way with an exponential security parameter $t(\ell) = 2^{\ell^{\Omega(1)}}$ then the function $AH_h(\ell, k)$ is an exponential function of ℓ for any fixed $k \geq 1$.

Now we are ready to state the reduction.

Theorem 3.2. *Assume that f is an $NP \cap coNP$ function with unique witnesses and $c \geq 1$ is a constant. Then any circuits computing moves of a Student solving (T) in c steps must have total size at least*

$$AH_f(n^{1/3}, 4c) - (c-1) \cdot n^{O(1)}.$$

The $O(1)$ constant depends on f but not on c.

In particular, if $AH_f(\ell, k)$ is a super-polynomial function of ℓ for any fixed $k \geq 1$, then the hypothesis (H) holds for f, and if $AH_f(\ell, k)$ is exponential for any fixed $k \geq 1$ then the total size of the circuits computing Student's moves have to be exponential in n.

Proof. Let f be a function satisfying the hypothesis of the theorem. Assume that for some constant $c \geq 1$ and an n large enough, a Student computed by circuits of the total size s solves the task (T) in c steps. We want to derive a lower bound on s.

Assume that for a given $x \in \{0,1\}^n$ the communication between the Student and the Teacher stops after the kth step of the Student, his candidate solutions in the computation being $i_1, \ldots, i_k$ (and i_k is correct). Call the k-tuple $(i_1, \ldots, i_k)$ the *trace of the computation* on x and denote it $Tr(x)$. Note that $k \leq c$ and that the trace determines also the Teacher's messages because of the assumption of the unique witnesses for f.

Claim 1. *There is a k-tuple $\overline{i} = (i_1, \ldots, i_k) \in [m]^k$ for some $k \leq c$ that is the trace of computations on at least a fraction of $\frac{2}{(3m)^k}$ of all inputs x.*

To prove the claim construct by induction on t a string $(i_1, \ldots, i_t) \in [m]^t$ such that the traces of at least $\frac{1}{3^{t-1}m^t}$ of all inputs x start with the t-tuple. For $t = 1$ note that there are m possible values and hence at least one of them, say i_1, appears at the beginning of at least a fraction of $1/m$ traces of all inputs. For the induction step assume we have a t-tuple $(i_1, \ldots, i_t)$ with the required property and consider two cases: (i) $(i_1, \ldots, i_t)$ is actually the whole trace of at least $2/3$ of all inputs whose traces start with $(i_1, \ldots, i_t)$, and (ii) otherwise.

In case (i) $(i_1, \ldots, i_t)$ is the required trace. In case (ii) extend $(i_1, \ldots, i_t)$ by i_{t+1} so that for at least $1/(3m)$ of all inputs with traces starting with $(i_1, \ldots, i_t)$ the traces will start with $(i_1, \ldots, i_{t+1})$ as well. This is possible because in case (ii) at least a third of all computations with traces starting with $(i_1, \ldots, i_t)$ continue and there are m choices for i_{t+1}. This proves the claim.

For the rest of the proof fix a trace $\overline{i} = (i_1, \ldots, i_k)$ provided by the claim. For $u \in \{0,1\}^\ell$ and $v \in \{0,1\}^{n-\ell}$ define $w(u,v) \in \{0,1\}^n$ by putting bits of u into bits of w in positions J_{i_k} (in the natural order) and then fill the remaining $n - \ell$ bits of w by bits of v (again in the natural order). An averaging argument yields the following claim.

Claim 2. *There is an $n - \ell$-tuple $a \in \{0,1\}^{n-\ell}$ such that there is at least a fraction $\frac{1}{(3m)^k}$ more $u \in \{0,1\}^\ell$ with $Tr(w(u,a)) = \overline{i}$ than those with $Tr(w(u,a))$ properly containing $\overline{i}$.*

Fix one such an $(n - \ell)$-tuple a. Because matrix A_n is a (d,ℓ)-design, for any row $i \neq i_k$ at most $d = \log(n+1)$ input bits from J_i are not set by a. Hence there are at most $n+1$ assignments v to bits in J_i not set by a. For each such v let z_v be the (unique) witness for the value of f on the assignment given by v and a to variables in J_i, and let Y_i be the set of all these witnesses z_v. Note that the total bit size of each Y_i is $(n+1)\ell^{O(1)} = n^{O(1)}$, and that there are n of them.

Now we define an algorithm C that attempts to compute f on inputs of length ℓ, and uses $\overline{i}$, a and all Y_i's as an advice. The algorithm will invoke the Student (i.e. the circuits computing its moves) and this is an additional source of non-uniformity of C.

Upon receiving an input $u \in \{0,1\}^\ell$ C defines the string $w := w(u,a) \in \{0,1\}^n$ and starts computation as the Student on $x := w$. Let U be those inputs u for which the trace $Tr(w(u,a))$ is either $\overline{i}$ or starts with $\overline{i}$, and let V be the complement of U. Define b_0 to be the majority value of f on V.

If the Student's first candidate solution is different from i_1 then C halts and produces b_0 as the output $C(u)$. (If C had a source of random bits then it would output a random bit at this point but deterministic C needs a fixed value.) If the first candidate solution is i_1 algorithm C reads from Y_{i_1} the right witness y_1 and sends it to the Student in place of the Teacher. Note that the uniqueness of witnesses for f implies that there is exactly one suitable string in Y_{i_1} and that C can find it in polynomial time: the size of Y_{i_1} is polynomial and each string can be tested in polynomial time.

In an analogous manner, if any of the candidate solutions the Student produces in steps $1, \ldots, k-1$ is different from the particular i_j, $j = 1, \ldots, k-1$, C halts and outputs the value b_0. Otherwise C sends to the Student always the correct witness it reads in the appropriate Y_i's. If the computation halts before reaching the kth step C again outputs b_0.

Finally we reach the kth step. If the Student's candidate solution is different from i_k C outputs b_0. But if it is i_k it outputs $1 - b_{i_k}$.

Claim 3. *The algorithm C computes correctly f on at least a fraction of*

$$1/2 + \frac{1}{(3m)^k} \geq 1/2 + \frac{1}{(3(\ell^3+1)^c)} \geq 1/2 + \frac{1}{\ell^{4c}}$$

of all inputs $u \in \{0,1\}^\ell$.

The algorithm outputs the bit b_0 in all cases except when the computation reaches the kth step and the Student produces i_k as its candidate solution. If the

Student/Teacher computation actually stops at that point, then the value $1 - b_{i_k}$ in indeed equal to $f(u)$. If the computation were to continue then we have no information. But note that by the choice of a in Claim 2 the former case happens for at least a fraction $\frac{1}{(3m)^k}$ more inputs $u \in \{0,1\}^\ell$ than the latter case. Because b_0 is the correct value of f for at least half of $u \in V$, the overall advantage algorithm C has in computing f is at least $\frac{1}{(3m)^k}$.

Claim 4. *The algorithm C can be computed by a circuit of size at most $s + (c - 1) \cdot n^{O(1)}$.*

C proceeds as the Student except when it needs to simulate the Teacher and find an appropriate witness in one of the sets Y_i. This is done at most $(c-1)$-times and takes $n^{O(1)}$ time each.

Claims 3 and 4 imply that

$$AH_f(n^{1/3}, 4c) \leq s + (c-1) \cdot n^{O(1)}$$

and the theorem follows. □

Lemma 3.1 and Theorem 3.2 imply the following corollary.

Corollary 3.3. *Assume that there exists a polynomial time permutation h such that for any fixed $k \geq 1$ it is ℓ^{-k} one way with a super-polynomial security parameter $t(\ell) = \ell^{\omega(1)}$. Then the hypothesis (H) holds.*

If the permutation h is ℓ^{-k} one way with even an exponential security parameter $t(\ell) = 2^{\ell^{\Omega(1)}}$ for all fixed $k \geq 1$, then the hypothesis (H) holds even when asserting that the total size of the circuits computing Student's moves have to be exponential in n.

We shall conclude this section with two remarks useful for a later reference.

Remarks.

(a) We have chosen the value $m := n+1$ in order to maximize a time bound in Sec. 4 (ideally this would translate into lower bounds for lengths of proofs). However, the construction allows bigger values of m, up to exponential in ℓ. First, m influences the rate of the advantage the algorithm C has in computing f: when m is polynomial in ℓ the advantage is polynomially small $\ell^{-O(1)}$, while for an exponential m it would be exponentially small $2^{-\ell^{\Omega(1)}}$. It is consistent with the present knowledge that there are NP ∩ coNP functions which is difficult to approximate even with an exponentially small advantage but it is clearly a stronger assumption to make than the standard hypothesis about one way permutations we have used.

Second, m does not appear explicitly in the size estimate for C but it is involved implicitly: the Student knows b which has size m. Hence for values of m that are super-polynomial in n one should estimate the size of C in terms of m rather than in terms of n, and it would be exponential in ℓ if m was.

(b) Assume we would want to allow f not only from $\mathrm{NP} \cap \mathrm{coNP}$ but from a larger class $NTime(r(\ell)) \cap coNTime(r(\ell))$ with a super-polynomial $r(\ell)$. In such a case the witnesses for the function values will have the length $O(r(\ell))$.

The size of witnesses plays a role in the estimate of the size of C: each step when C simulates the Teacher and searches for a witness for $f(x(J_i)) = b_i$ in Y_i would now take time $O(nr(\ell)^{O(1)})$. Hence the size of C would be estimated by $s + O(cnr(\ell)^{O(1)}) = s + O(c\ell^3 r(\ell)^{O(1)})$. For this to give a non-trivial upper bound one must have $r(\ell) \leq 2^{(1-\Omega(1))\ell}$ or better still $r(\ell) \leq 2^{\ell^{\Omega(1)}}$. In particular, one cannot allow f from $NE \cap coNE$. This is relevant to note because a version of the Razborov's conjecture allowing $NE \cap coNE$ in place of $\mathrm{NP} \cap \mathrm{coNP}$ was shown to have a startling consequence in [17].

4. A Model of T_{PV} Where Task (T) has no Solution

The language L_{PV} of Cook's [6] theory PV has a name for every polynomial-time algorithm obtained from some basic algorithms by the composition and by the limited recursion on notation, following Cobham's [5] characterization of polynomial time. The axioms of PV codify how the algorithms are built from each other. Cook has originally defined PV as an equational theory but after [22] it became customary to define it as a usual first-order theory whose axioms are purely universal statements; this is the convention we adopt here. We do not spell out the definition of either L_{PV} or PV as we use only two facts: every function symbol from L_{PV} is in the standard model $\mathbf{N}$ interpreted by a polynomial time function and PV is true in the standard model ([13] offers all details).

Theory PV is closely linked with Extended Frege system EF and this relation has many facets. For example, PV proves the soundness of EF and EF can simulate by polynomial size proofs of instances any PV proof of a universal statement. We shall, however, not use this relation (at least not directly). In fact, we shall work instead with the true universal first-order theory of $\mathbf{N}$ in the language L_{PV}. We shall denote this theory T_{PV}. Note that T_{PV} contains formulas expressing the soundness of all proof systems and that $\mathrm{PV} \subseteq \mathrm{T}_{\mathrm{PV}}$.

A class of models of T_{PV} can be constructed as follows. Let M be a non-standard model of true arithmetic (tacitly in the language L_{PV}). Let $n \in M$ be a non-standard number and define M_n and M_n^* to be the substructures of M consisting of numbers whose bit length is less than n^k for some standard $k \in \mathbf{N}$ or less than $2^{n^{1/k}}$ for all standard $k \in \mathbf{N}$, respectively (such models are called a small and a large canonical model in [15]). Both M_n and M_n^* are indeed closed under all functions in the language L_{PV} as these have polynomial length-growth. Note that, for example, matrix A_n is in $M_n \subseteq M_n^*$.

Theorem 4.1. *Assume f is an $NP \cap coNP$ function with unique witnesses. Let M be a non-standard model of true arithmetic, $n \in M$ a non-standard element, and let $b \in M$ be any string of length m $(= n + 1)$ that is outside of the range of $NW_{A_n,f}$ (such a string b exists as the domain of $NW_{A_n,f}$ is smaller than 2^m).*

If, for all fixed $k \geq 1$, $AH_f(\ell, k)$ is a super-polynomial function of ℓ, then there exists a model N of T_{PV} that is a cofinal extension of M_n, and a string $w \in N$ of length n such that in N it holds:

$$\forall\, i \in [m] f(w(J_i)) = b_i. \tag{4.1}$$

If, for all fixed $k \geq 1$, $AH_f(\ell, k)$ is an exponential function of ℓ, then there exists a model N^* of T_{PV} that is a cofinal extension of M_n^*, and a string $w \in N^*$ of length n such that in N^* (4.1) holds.

Proof. Let T be a theory consisting of the diagram of M_n (with a name for every element of M_n and, in particular, for A_n and b) together with the theory T_{PV}. Assume for the sake of contradiction that no model N with the required properties exists (as the statement in question is bounded to elements whose length is bounded by elements of M_n, we may consider without a loss of generality only cofinal extensions of M_n).

By completeness of first-order logic, it follows that T proves

$$\forall\, x(|x| = n)\exists\, i \in [m]\forall\, y(|y| \leq n^{O(1)})\ \neg F_{b_i}(x(J_i), y)$$

where F_0 and F_1 are the polynomial-time relations defining f as in (2.2), and are represented here by open formulas of L_{PV}.

As T is a universal theory we may apply the KPT witnessing theorem from [22] (see also [13, 7]), a form of Herbrand theorem, and conclude that there are terms of the language of T

$$i_1(x), i_2(x, y_1), \ldots, i_c(x, y_1, \ldots, y_{c-1}) \tag{4.2}$$

for some natural number $c \geq 1$, such that each term i_k depends only on the variables shown and may use constants from M_n, and such that T proves (and hence in M_n it holds) the universal closure of the following disjunction:

$$\begin{gathered}(i_1 \in [m] \wedge \neg F_{b_{i_1}}(x(J_{i_1}), y_1)) \vee \\ (i_2 \in [m] \wedge \neg F_{b_{i_2}}(x(J_{i_2}), y_2)) \vee \\ \cdots \\ (i_c \in [m] \wedge \neg F_{b_{i_c}}(x(J_{i_c}), y_c)).\end{gathered} \tag{4.3}$$

(We have left out the bounds to the lengths from the formula.)

The value of each term i_j can be computed in time polynomial in n as they contain only parameters of polynomial length, functions interpreting the language are polynomial-time and the input length is polynomially bounded too.

It remains to observe (as it is now standard) that this disjunction defines an algorithm for a Student that solves the task (T) in c steps: The Student first proposes solution $i_1(x)$. If it fails it proposes $i_2(x, y_1)$ computed from x and witness y_1 to $f(x(J_{i_1})) = b_{i_1}$ provided by the Teacher, etc. The fact that the disjunction is universally valid means that the Student must find a correct solution in at most c steps.

Hence this gives a polynomial $n^{O(1)}$ time non-uniform algorithm for the Student which contradicts Theorem 3.2.

The case of M_n^* is completely analogous except that parameters in the terms are now of sub-exponential size $2^{n^{o(1)}}$ and this yields also a sub-exponential time bound for the Student. □

At the beginning of Sec. 2, we have set $m = n + 1$. If the function f has exponential approximating hardness we get by the previous theorem $N^* \supseteq M_n^*$ and this model contains strings of length up to anything subexponential $2^{n^{o(1)}}$. This translates in the next statement (part 2) to the a bigger time allowed for a non-deterministic algorithm defining set R: it can be also subexponential $2^{\ell^{o(1)}}$. But if f has even exponential hardness $H_f(\ell)$ we can take also a bigger value for the parameter m, any $m = m(n) \leq 2^{n^{o(1)}}$. Now the time bounds allowed for definitions of R, as measured in terms of m, include all polynomial bounds (and can include slightly super-polynomial ones) but not arbitrary subexponential.

Theorem 4.2 (Consistency of Statement (S)). *Let $k \geq 1$ be a natural number parameter and $m = m(k)$ an injective function of k, and assume $k < m(k) < 2^{k^{o(1)}}$. Let A_k be $k \times m$ 0-1 matrices that are $(\log m, k^{1/3})$ designs. Assume f is an NP ∩ coNP function with unique witnesses. Then the following three statements hold:*

(1) *If $m = n + 1$ and, for all fixed $k \geq 1$, $AH_f(\ell, k)$ is a super-polynomial function of ℓ, and R is an infinite NP set then it is consistent with T_{PV} that*

$$R \cap \mathrm{Rng}(g) \neq \emptyset.$$

(2) *If $m = n + 1$ and, for all fixed $k \geq 1$, $AH_f(\ell, k)$ is an exponential function of ℓ, and R is an infinite set from the class $NTime(2^{m^{o(1)}})$ then it is consistent with T_{PV} that*

$$R \cap \mathrm{Rng}(g) \neq \emptyset.$$

(3) *Let $m(k)$ be any function that is $2^{k^{o(1)}}$ and assume that $H_f(\ell)$ is an exponential function of ℓ. Assume R is an infinite NP set that has infinitely many elements whose length equals to $m(k)$ for some $k \geq 1$.*

Then it is consistent with T_{PV} that

$$R \cap \mathrm{Rng}(g) \neq \emptyset.$$

Proof. Let us start with the first statement; the proof of the second is completely analogous using the remark before the theorem.

Let R be an infinite NP set and assume that it is defined by the condition $\exists y(|y| \leq |x|^{O(1)}) R_0(x, y)$, where R_0 p-time.

We may assume that R is disjoint with $\mathrm{Rng}(g)$ as otherwise there is nothing to prove. Let M be a non-standard model of true arithmetic. Set R has non-standard elements in any such model; take one such element from M and denote it b. It is not in $\mathrm{Rng}(g)$.

The fact that $b \in R$ is witnessed by some polynomially longer string c such that $R_0(b, c)$ holds.

Now define $m = |b|$ and $n = m - 1$ and apply Theorem 4.1 to get a model $N \supseteq M_n$ of $\mathrm{T_{PV}}$ in which b is in the range of g. But c is also in N and $R_0(b, c)$ holds, so $b \in R$ in N too.

The proof of the third statement is again analogous but we need to use the extra lengths-condition posed on R to guarantee that b has the length of the form $m(n)$, and also the assumption that $H_f(\ell)$ is exponential (approximating hardness would not suffice) because the advantage the algorithm in Theorem 3.2 gets in computing f is $\Omega(m^{-c})$ which is only $2^{-\ell^{o(1)}}$ if $m = 2^{\ell^{o(1)}}$. □

Statement (S) is related to proof complexity as follows. Denote (S1), (S2) and (S3) the three statements whose consistency was established in Theorem 4.2; all three have the form[a] that, under certain assumptions on map g and set R, $R \cap \mathrm{Rng}(G) \neq \emptyset$.

Let P be any proof system and $r \geq 1$ a constant, and let $m(k) = k + 1$ and f have super polynomial approximating hardness as in (S1). Because the length of the τ-formula $\tau(g)_b$ is polynomial in the length of b, the set

$$R_P := \{b \in \{0,1\}^* \mid \tau(g)_b \text{ has a } P\text{-proof of size at most } |b|^r\}$$

is in NP, and by the soundness of P it is disjoint with $\mathrm{Rng}(g)$. Statement (S1) therefore implies that g is hard for P.

If f has an exponential approximating hardness and $m(k) = k + 1$ then we get analogously an exponential lower bound on the P-proofs of the formulas $\tau(g)_b$, via (S2).

If f has an exponential hardness then $m(k)$ can be any subexponential function, and (S3) analogously implies that g is hard for P. In fact, we can get m up to 2^{k^δ} for some $\delta > 0$ by compactness. These are the parameters in Razborov's conjecture.

From the proof complexity point of view the best choice of $m(k)$ is indeed $k+1$ as it translates into better lower bounds.

5. Missing Reflection

Let P be a proof system and R_P be the NP set from the end of the last section. If R_P is finite for all $r \geq 1$ then g is hard for P and thus P is not p-bounded.

If R_P is infinite take the model N from Theorem 4.1 and the string from M_n satisfying in N $b \in \mathrm{Rng}(g) \cap R_P$. The formula $\tau(g)_b$ has a P-proof in M_n (a witness to the membership of b in R_P) and hence also in N. The soundness of P is a true universal statement and hence valid in N. But in N the statement the τ-formula

[a] These statements seem also akin in form to the demi-bit conjecture of Rudich [28].

encodes is false. That should yield a contradiction and hence a proof that R_P is finite, and as $r \geq 1$ was arbitrary also that g is hard for P. This type of a lower bound argument goes back to Ajtai [1].

Unfortunately the soundness of P, the usual reflection principle, that is valid in N is too weak to support this reasoning. That principle says that if a formula has a P-proof then no truth assignments can falsify it. The formula $\tau(g)_b$ has the form

$$\bigvee_{i \in [m]} \psi_i(x, y^i)$$

where $\psi_i(x, y^i)$ is a propositional translation (with respect to the length $|x| = n$) of the formula

$$|y^i| \leq n^{O(1)} \to \neg F_{b_i}(x(J_i), y^i)$$

saying that no y^i is a witness that $f(x(J_i)) = b_i$.

Let π be its P-proof from M_n (i.e. of polynomial size). Substituting (inside N) for $x := w$ (without loss of generality we may assume that P has such a substitution property) we get in N a P-proof σ of

$$\bigvee_{i \in [m]} \psi_i(w, y^i).$$

What we have in model N are truth assignments v^i falsifying each $\psi_i(w, y^i)$ individually but we do not have there one string $v = (v^1, \ldots, v^m)$ collecting all v^i together and providing a truth assignment for the whole disjunction.

The existence of such a string v can be deduced from the existence of individual strings v^i via the so called sharply bounded collection scheme (cf. [4, 13]). Its instance looks like:

$$\forall\, i \exists\, y^i B(i, y^i) \to \exists\, z \forall\, i B(i, (z)_i)$$

where i are sharply bounded (by m in our case), y^i and z are bounded, and B is an open PV-formula. Unfortunately, Cook and Thapen have proved that this scheme is not provable in $\mathrm{T_{PV}}$ unless factoring is not hard, cf. [9].[b]

Let us introduce a handy notation:

$$\dot{\bigvee}_i \ \varphi_i$$

meaning that the formulas φ_i in the disjunction use disjoint sets of variables, i.e. no two formulas share a single variable.

Define the *disjunction soundness* of P to be the following principle:

- *If P proves a formula of the form $\dot{\bigvee}_i \varphi_i$, then at least one of φ_i is a tautology.*

[b] In fact, it would be enough to arrange that the instance of the collection scheme needed holds in some extension of N only. But Thapen pointed out that the argument of [9] should rule out this option too.

If we could arrange the disjunction soundness in N then the argument outlined above would yield Statement (S) and hence Razborov's conjecture.

For a proof system P we shall say that P admits *feasible disjunction property* (FDP) if:

- *There is $c \geq 1$ such that whenever π is a P-proof of a disjunction $\dot{\bigvee}_i \varphi_i$ then there is a P-proof σ of one of the disjuncts φ_i and of the size $|\sigma| \leq |\pi|^c$.*

A simple but maybe useful observation is that for the purpose of proving that P is not p-bounded (an arbitrary P) we may assume without a loss of generality that P does admit the feasible disjunction property. This is because the failure of the property automatically implies that P is not p-bounded: one of the disjuncts φ_i must be a tautology and its P-proofs cannot be p-bounded.

By the observation we may assume that P has the FDP in M_n. If we had the FDP for P in N too we could get the wanted contradiction: one of the formulas $\psi_i(w, y^i)$ would have a P-proof in N and we have falsifying assignments for all of them (and the ordinary soundness applies). Note that we cannot hope to prove FDP for strong P in PV as that would entail (via a witnessing argument) feasible interpolation for P and that is known to contradict the security of RSA, see [20].

In fact, similarly as it would suffice to have an extension of N with a witness to the collection scheme, also the FDP property can be weakened: it would suffice that for some $i \in [m]$ there is an extension of N in which $\psi_i(w, y^i)$ has a P-proof. It is perhaps worthwhile to point out that this property does hold for disjunctions of two formulas.

6. Specializations to Smaller Computational Classes

We may specialize the whole situation to some smaller classes of circuits. Of particular interest from the point of view of proof complexity are AC^0, $AC^0(q)$, and NC^1. Here we shall comment on the first two classes as they offer a possibility to perform the construction without an unproven assumption or with a significantly weaker one.[c]

Specializing to a circuit class means to consider NP ∩ coNP functions f defined as in (2.2) but with formulas F_a in the class, and allowing also functions $S_1, \ldots, S_c$ in (2.3) defining the Student's moves only from the class. Note then that the construction of circuit C will yield again a circuit in the same class. One only needs to verify that the steps when C simulates the Teacher and searches for a witness can be performed in a constant depth. A simple definition by cases

$$(y_i)_j := \bigwedge_{z \in Y_i} (F_{b_i}(u, z) \wedge (z)_j)$$

expressing the jth bit of the correct witness y_i works.

[c] The case of NC^1 is being worked out by J. Pich (in preparation). It relates to another Razborov's conjecture from [27].

It is well-known that in the AC^0 case the random restriction method (in the sharp version of [11]) shows that, for any fixed d, the parity function needs an exponential size depth d circuit even for an approximation with an exponentially small advantage (cf. [3]). This readily yields a suitable $\mathrm{NP} \cap \mathrm{coNP}$ function with unique witnesses and our argument then proves the AC^0 version of the hypothesis (H) unconditionally (even with an exponential lower bound for the Student).

For the class $AC^0(q)$ the situation is different. For q a prime the Razborov–Smolensky approximation method [26, 29] allows to approximate any small $AC^0(q)$ circuit by a low degree polynomial over $\mathbf{F}_q$ with a very small error. For example, setting the parameters right allows to approximate any polynomial size circuit by a degree d polynomial with a negligible error, as long as $d = \omega(\log \ell)$.

Hence finding a suitable $\mathrm{NP} \cap \mathrm{coNP}$ function f that needs super-polynomial size $AC^0(q)$ circuit for an approximation with any particular $\ell^{-O(1)}$ advantage is equivalent to finding f that cannot be approximated well by a low degree polynomial over $\mathbf{F}_q$. The qualification "low" here means at least $\omega(\log \ell)$ (and $\ell^{\Omega(1)}$ would yield exponential lower bounds).

Unfortunately this problem is open. In fact, it is even open if there is an NP function which no degree $\log \ell$ polynomial approximates with an advantage $1/\ell$. See [30] for a survey of this topic.

Acknowledgments

I am indebted to N. Thapen (Prague) for comments on the preliminary version of the paper.

Supported in part by grants IAA100190902, MSM0021620839, LC505 (Eduard Čech Center) and by a grant from the John Templeton Foundation. Also partially affiliated with the Institute of Mathematics of the Academy of Sciences and grant AV0Z10190503.

References

[1] M. Ajtai, The complexity of the pigeonhole principle, in *Proc. IEEE 29th Annual Symp. on Foundation of Computer Science* (1988), pp. 346–355.

[2] M. Alekhnovich, E. Ben-Sasson, A. A. Razborov and A. Wigderson, Pseudorandom generators in propositional proof complexity, *Electronic Colloquium on Computational Complexity*, Rep. No. 23 (2000). Ext. abstract in *Proc. of the 41st Annual Symp. on Foundation of Computer Science* (2000), pp. 43–53.

[3] S. Arora and B. Barak, *Computational Complexity: A Modern Approach* (Cambridge University Press, 2009).

[4] S. R. Buss, *Bounded Arithmetic* (Naples, Bibliopolis, 1986).

[5] A. Cobham, The intrinsic computational difficulty of functions, in *Proc. Logic, Methodology and Philosophy of Science*, ed. Y. Bar-Hillel (North-Holland, 1965), pp. 24–30.

[6] S. A. Cook, Feasibly constructive proofs and the propositional calculus, in *Proc. 7th Annual ACM Symp. on Theory of Computing* (ACM Press, 1975), pp. 83–97.

[7] S. A. Cook and P. Nguyen, *Logical Foundations of Proof Complexity* (Cambridge University Press, 2010).

[8] S. A. Cook and Reckhow, The relative efficiency of propositional proof systems, *J. Symb. Log.* **44**(1) (1979) 36–50.

[9] S. A. Cook and N. Thapen, The strength of replacement in weak arithmetic, *ACM Trans. Comput. Log.* **7**(4) (2006).

[10] O. Goldreich, *Foundations of Cryptography*, Vol. 1 (Cambridge University Press, 2001).

[11] J. Hastad, Almost optimal lower bounds for small depth circuits, in *Randomness and Computation*, ed. S. Micali, Advances in Computing Research, Vol. 5 (JAI Press, 1989), pp. 143–170.

[12] E. Jeřábek, Dual weak pigeonhole principle, Boolean complexity, and derandomization, *Ann. Pure Appl. Log.* **129** (2004) 1–37.

[13] J. Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, Encyclopedia of Mathematics and Its Applications, Vol. 60 (Cambridge University Press, 1995).

[14] J. Krajíček, On the weak pigeonhole principle, *Fund. Math.* **170**(1–3) (2001) 123–140.

[15] J. Krajíček, Tautologies from pseudo-random generators, *Bull. Symb. Log.* **7**(2) (2001) 197–212.

[16] J. Krajíček, Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds, *J. Symb. Log.* **69**(1) (2004) 265–286.

[17] J. Krajíček, Diagonalization in proof complexity, *Fund. Math.* **182** (2004) 181–192.

[18] J. Krajíček, *Forcing with Random Variables in Bounded Arithmetic, and proof Complexity*, London Mathematical Society Lecture Notes Series, Vol. 382 (Cambridge University Press, 2011).

[19] J. Krajíček and P. Pudlák, Propositional proof systems, the consistency of first order theories and the complexity of computations, *J. Symb. Log.* **54**(3) (1989) 1063–1079.

[20] J. Krajíček and P. Pudlák, Some consequences of cryptographical conjectures for S^1_2 and EF, *Inform. Comput.* **140**(1) (1998) 82–94.

[21] J. Krajíček, P. Pudlák and J. Sgall, Interactive computations of optimal solutions, in *Mathematical Foundations of Computer Science* (B. Bystrica, August '90), ed. B. Rovan, Lecture Notes in Computer Science, Vol. 452 (Springer-Verlag, 1990), pp. 48–60.

[22] J. Krajíček, P. Pudlák and G. Takeuti, Bounded arithmetic and the polynomial hierarchy, *Ann. Pure Appl. Log.* **52** (1991) 143–153.

[23] N. Nisan and A. Wigderson, Hardness vs. randomness, *J. Comput. Syst. Sci.* **49** (1994) 149–167.

[24] J. Pich, Nisan-Wigderson generators in proof systems with forms of interpolation, *Math. Log. Quart.* **57**(4) (2011) 379–383.

[25] P. Pudlák, The lengths of proofs, in *Handbook of Proof Theory*, ed. S. R. Buss (Elsevier, 1998), pp. 547–637.

[26] A. A. Razborov, Lower bounds on the size of bounded depth networks over a complete basis with logical addition, *Mat. Z.* **41**(4) (1987) 598–607.

[27] A. A. Razborov, Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution, preprint (2003).

[28] S. Rudich, Super-bits, demi-bits, and $\tilde{NP}/qpoly$-natural proofs, in *Proc. of the 1st Int. Symp. on Randomization and Approximation Techniques in Computer Science*, Lecture Notes in Computer Science (Springer-Verlag, Vol. 1269, 1997), pp. 85–93.

[29] R. Smolensky, Algebraic methods in the theory of lower bounds for Boolean circuit complexity, in *Proc. 19th Ann. ACM Symp. on Th. of Computing* (1987), pp. 77–82.

[30] E. Viola, Correlation bounds for polynomials over $\{0, 1\}$, *ACM SIGACT News* **40**(1) (2009).