Unprovability of circuit upper bounds in Cook's theory PV

Jan Krajíček Igor C. Oliveira*

Faculty of Mathematics and Physics
Charles University in Prague

Abstract

We establish unconditionally that for every integer $k \geq 1$ there is a language $L \in \mathrm{P}$ such that it is consistent with Cook's theory PV that $L \notin \mathrm{SIZE}(n^k)$. Our argument is non-constructive and does not provide an explicit description of this language.

1 Introduction

Bounded arithmetic theories constitute a class of weak subtheories of Peano arithmetic with close ties to computational complexity theory. Prominent among them is theory PV defined by Cook [5] as an equational theory and later reformulated as a universal first order theory in [12, 11].

Theory PV or its mild extensions seem to formalize most of contemporary complexity theory (cf. [13, 3, 11, 8, 9, 10, 7, 14, 15, 16] and references therein). For instance, it is known that the PCP Theorem can be formalized and proved in PV [16]. It is thus of interest to understand, given an established conjecture, whether it is provable in one of these theories or at least consistent with them.

An unprovability statement can be understood as a result illustrating the inadequacy of methods available in the respective theory. This is studied in complexity theory as the so called barriers (cf. [2, 18, 1]), often formulated using ad hoc concepts hard to compare with each other. The unprovability results on the other hand are in the tradition how mathematical logic captured (and answered) similar questions in other parts of mathematics.

The latter direction, to show the consistency of the conjecture in question with PV or with stronger theories, is at least as interesting as showing its unprovability. Such a consistency result says that, although we do not know if the conjecture is true (meaning true in the standard model of natural numbers), we know that it is true in a non-standard model of a theory so strong that complexity theory looks in it almost indistinguishable from the standard one.

In this work we study the provability of circuit upper bounds (or equivalently, the consistency of lower bounds). Circuit lower bounds were considered in bounded

*Supported in part by CNPq grant 200252/2015-1.

arithmetic by Razborov [17] in a particular formalism. We use the somewhat more intrinsic formalism of [13, 3, 11] and followed in [8, 6, 7, 14, 15, 16].

It has been proved in [6], assuming that NP $\not\subseteq$ coNP$/O(1)$ or that the polynomial time hierarchy does not collapse to the Boolean hierarchy, that it is consistent with PV that NP $\not\subseteq$ P/poly. Here we prove *unconditionally* that for every $k \geq 1$ there is a language $L \in \mathrm{P}$ such that it is consistent with Cook's theory PV that $L \notin \mathrm{SIZE}(n^k)$, where $\mathrm{SIZE}(n^k)$ denotes the class of languages decided by *non-uniform* Boolean circuits of size at most $O(n^k)$. We refer to the statement of Theorem 1 below for a precise formulation of the result.

We do not know how to extend our result to Buss's theory S^1_2 from [3] (results from [6] were extended there to S^1_2) or how to show that one can take SAT for L for all $k \geq 1$. Perhaps the most accessible problem is to extend our result to PV augmented by the dual weak pigeonhole principle for polynomial time functions, a theory denoted APC_1 by some authors.

2 Formalization and statement of the theorem

The language of PV has function symbols for all polynomial time algorithms as generated by Cobham's limited recursion on notation [4]. All axioms of PV are universal formulas codifying how particular algorithms are defined from each other. The details of the definition of PV are fairly technical, but such details are needed only for establishing links between PV and propositional proof systems (cf. [11]). We use a form of Herbrand's Theorem (see below), and for that it only matters that the axioms are universal formulas. In fact, we could add to PV any set of true universal sentences as additional axioms, and our unprovability result would still hold.

We will talk about polynomial time algorithms in the theory meaning that they are represented by the corresponding function symbols. We shall claim on a few occasions that some algorithm f_1 constructed in a particular way from another algorithm f_2 can be defined in PV; this means that PV proves that f_1 behaves as described in the definition. In all cases this is straightforward but tedious, and presupposes a certain amount of bootstrapping of PV which is part of standard background in bounded arithmetic (see e.g. in [3] how this is done). The details are not necessary for understanding our argument and can be found in [3, 5, 11, 12].

For a unary PV function symbol f and integers $k, c \geq 1$, denote by $\mathrm{UP}_{k,c}(f)$ the sentence

$$\forall 1^{(n)} \exists \text{circuit } C_n(|C_n| \leq cn^k) \forall x(|x| = n),\ f(x) \neq 0 \leftrightarrow C_n(x) = 1\ , \tag{1}$$

which asserts that the (polynomial time) language defined by f admits a (non-uniform) sequence of circuits of size at most cn^k.[1] (We refer to [11, 14] for more information about the formalization of circuit complexity in bounded arithmetic.)

Theorem 1. *For every $k \geq 1$ there is a unary PV function symbol h such that for no constant $c \geq 1$ PV proves the sentence* $\mathrm{UP}_{k,c}(h)$.

[1]For the reader familiar with bounded arithmetic, we stress that we abuse notation and use $|C_n|$ to denote the number of gates in C_n, while $|x|$ refers to the length of x in the usual sense. Also, the symbol $\forall 1^{(n)}$ abbreviates the universal quantification over strings of the form $1^{(n)}$, i.e., strings consisting of a sequence of ones.

The high level idea of the proof is: (i) the provability of (1) implies a certain uniformity of the family of circuits, and (ii) we can adapt the proof by Santhanam and Williams from [19] that P has no uniform sequences of circuits of size $O(n^k)$, for any fixed $k \geq 1$. Complications arise as the uniformity given by (i) is more general than the one employed in (ii). In particular, it is not clear how to establish Theorem 1 using only the soundness of PV and (extensions of) the Santhanam-Williams Theorem.

To get around this difficulty we argue roughly as follows. Either a candidate sentence $\mathrm{UP}_{k,c}(g)$ that we start with is not provable in PV (and we are done), or we extract from any proof of this sentence a finite number of languages in P such that PV cannot prove that all of them admit circuits of size $O(n^k)$. We remark that the non-constructive aspect of the result comes from the fact that the hard language and its deterministic time complexity may depend on a (possibly non-existent) proof of the initial sentence.

In order to implement this approach we use that PV is a universal theory for polynomial time computations, a formalization of the main ideas employed in the uniform circuit lower bound from [19], the KPT Theorem from bounded arithmetic (Theorem 2 below), and a finite number of recursive applications of Herbrand's Theorem. The argument has a few subtle points, and we make some additional observations after we present the proof of Theorem 1 in Section 4.

Remark 1. *An alternative and equally natural formalization of circuit upper bounds can be obtained via a single formula* $\mathrm{UP}_k(h)$ *that existentially quantifies over parameter* c*. This leads however to a sentence of higher quantifier complexity. While KPT witnessing* (*stated as Theorem 2 in Section 4*) *can be generalized in this direction, the information it then offers does not seem to yield the polynomial time algorithms our technique needs. In particular, we leave the unprovability of the modified version as an open problem.*

Theorem 1 and a standard compactness argument imply the following result.

Corollary 1. *For every* $k \geq 1$ *there exists a unary PV function symbol* h *and a model* $\mathfrak{M}_k$ *of PV such that for every* $c \geq 1$ *we have*

$$\mathfrak{M}_k \models \neg \mathrm{UP}_{k,c}(h).$$

In other words, from the point of view of $\mathfrak{M}_k$ there are languages in P that require non-uniform circuits of size $\omega(n^k)$.

3 Uniform sequences of circuits and PV

In this section we adapt a proof by Santhanam and Williams [19] that P is not included in (P-uniform)-SIZE(n^k). Here (P-uniform)-SIZE(n^k) is the class of languages recognizable by a polynomial time uniform family of circuits of size at most $O(n^k)$. That is, there is a polynomial time algorithm f that on input $1^{(n)}$ computes a description of a size cn^k circuit C_n, where $c \geq 1$ is a fixed constant. Following [19] we take as the description the set of all 4-tuples

$$(1^{(n)}, u, v, w) \tag{2}$$

where u, v are names of nodes ($\leq k(\log n + O(1))$ bits each) such that there is a wire from u to v, and w is the information about the type of the gate at v or about the input at v if v is an input node ($\leq \log n + O(1)$ bits). We assume that a special tuple indicates the output node of C_n. The language consisting of all 4-tuples (2) for all $n \geq 1$ is called L_{dc}, the direct connection language of $\{C_n\}_n$.

The following standard definitions play an important role in the argument. We use $\mathrm{DTIME}(n^d)/n^{2/3}$ to denote the class of languages recognizable by a time $O(n^d)$ algorithm with an advice of size $n^{2/3}$. We say that a language L is infinitely often in a complexity class Γ if L agrees on infinitely many input lengths with some language $L' \in \Gamma$.

The next lemma formalizes the deterministic time hierarchy theorem with a bounded amount of advice.

Lemma 1. *For every $d \geq 1$ there is $L \in DTIME(n^{d+1})$, represented by algorithm g_{d+1} computing its characteristic function, such that for every time $O(n^d)$ algorithm h working with advice $n^{2/3}$ there is $c_h \geq 1$ such that PV proves:*

$$\forall n \geq c_h \forall a(|a| = n^{2/3}) \exists x(|x| = n),\ h(x,a) \neq g_{d+1}(x)\ .$$

Proof (Sketch): The separation is reported as a folklore result in [19, Proposition 2.1]. We simply check that its proof formalizes in PV.

Define a time $O(n^{d+1})$ algorithm g_{d+1} that operates as follows. On an input x of length $|x| = n$:

- it interprets the first $\log n$ bits of x as a description of a time $n^d \log n$ algorithm h, and the next $n^{2/3}$ bits as advice a,
- runs h on x with advice a,
- outputs 0 if and only if the simulation ends with a non-zero value.

The constant $c_h \geq 8$ is chosen so that $\log c_h$ bits suffice to describe the particular h. Observe that in order for the sentence to hold for every large enough n it is important that the parts of the input corresponding to the description of the algorithm and the advice are disjoint.

q.e.d.

Take now $\{C_n\}_n$ a P-uniform sequence of size cn^k circuits and let f be the generating polynomial time algorithm. That is, on input $1^{(n)}$ f produces the list of 4-tuples as in (2). Following [19] we compress each such 4-tuple into the 5-tuple

$$(\mathrm{Bin}(n)01^{(n^{1/3k})}, u, v, w, 1^{(t)}) \tag{3}$$

where $\mathrm{Bin}(n)$ is the dyadic numeral for n (of length $\log n + O(1)$ bits) and t is chosen to pad the length of the 5-tuple to exactly $m(n) \stackrel{\mathrm{def}}{=} \lceil n^{1/(2k)} \rceil$ bits, as soon as n is sufficiently large (parameter t is not present in [19]). The language of all such 5-tuples obtained from L_{dc} is the language L_{succ}, the succinct version of L_{dc}. It is polynomial time and an algorithm $\tilde{f}$ recognizing it can be easily defined from f and, in particular, in PV.

Let $\mathrm{CircuitVal}(y, x)$ be the polynomial time algorithm evaluating circuit y on input x.

Lemma 2. *Let f, $\{C_n\}_n$, and $\tilde{f}$ be as above, and assume that for some $\tilde{c} \geq 1$*

$$\mathrm{PV} \ \vdash \ \mathrm{UP}_{k,\tilde{c}}(\tilde{f}) \ . \tag{4}$$

Let $g \stackrel{\text{def}}{=} g_{3k}$ for a fixed integer $k \geq 3$ be the function guaranteed to exist by Lemma 1. Then there exists $c_f \geq 1$ such that PV proves

$$\forall 1^{(n)}(n \geq c_f)\exists x(|x| = n), \ g(x) \neq C_n(x) \ , \tag{5}$$

where $C_n(x)$ abbreviates $\mathrm{CircuitVal}(f(1^{(n)}), x)$.

Proof: Our argument will follow the proof of [19, Theorem 1.1] and is done in PV. Assuming (5) fails we describe an explicit polynomial time algorithm h that will certify that g is (infinitely often) in $\mathrm{DTIME}(n^{3k-1})/n^{2/3}$. This contradicts the sentence from Lemma 1.

Algorithm h operates as follows. By the assumption (4) there are circuits D_m recognizing L_{succ} on m-bit inputs, where $m = m(n)$, as defined above. Upon receiving x, $|x| = n$, and advice string a, $|a| = n^{2/3}$, describing a candidate circuit D_m, h tries all possible 3-tuples (u, v, w) (among no more than $O(n^{2k+1})$ possibilities) and for each of them uses D_m to check if the corresponding 5-tuple as in (3) is in L_{succ}. Since for large enough n the corresponding circuit D_m has size $O(n^{1/2})$, this requires time $O(n)$ for each 5-tuple. There are $O(n^{2k+1})$ such simulations so the total time this part takes is $O(n^{2k+2})$.

After this stage h knows the description of C_n, a circuit of size at most cn^k, and uses it to compute a candidate value for $g(x)$ in time $O(n^{2k})$. Under our initial assumption, the algorithm is correct on infinitely many input lengths, which is contradictory if $k \geq 3$.

q.e.d.

Lemma 3. *Let f, g, k, $\{C_n\}_n$, and $\tilde{f}$ be as above, and assume that (4) holds. There is $c_f \geq 1$ and a polynomial time algorithm e such that PV proves*

$$\forall 1^{(n)}(n \geq c_f), \ |e(1^{(n)})| = n \wedge g(e(1^{(n)})) \neq C_n(e(1^{(n)})) \ . \tag{6}$$

That is, e provably produces witnesses to (5).

Proof: This follows from Lemma 2 using Herbrand's Theorem, as (5) is a $\forall\exists$-formula and PV is a universal theory.

q.e.d.

4 Proof of Theorem 1

We will need the following standard witnessing result from bounded arithmetic (the so called KPT theorem), stated below for convenience of the reader.

Theorem 2 ([12], see also [11])**.** *Let T be a universal theory with vocabulary $\mathcal{L}$, ϕ be an open $\mathcal{L}$-formula, and suppose that*

$$T \vdash \forall w \, \exists u \, \forall v \, \phi(w, u, v) \ .$$

Then there exist a constant $k \geq 1$ and a finite sequence $t_1, \ldots, t_k$ of $\mathcal{L}$-terms such that

$$T \vdash \phi(w, t_1(w), v_1) \vee \phi(w, t_2(w, v_1), v_2) \vee \ldots \vee \phi(w, t_k(w, v_1, \ldots, v_{k-1}), v_k) ,$$

where the notation $t_i(w, v_1, \ldots, v_{i-1})$ indicates that these are the only variables occurring in t_i.

We remark that Theorem 2 has a natural interpretation as an interactive game with finitely many rounds, and we refer to [15] for an example in the related context of circuit lower bounds.

Continuing with the proof of Theorem 1, assume

$$\mathrm{PV} \vdash \mathrm{UP}_{k,c}(g) , \tag{7}$$

where $g = g_{3k}$ and $c \geq 1$ is arbitrary. Observe that $\mathrm{UP}_{k,c}(\cdot)$ is a sentence of the form $\forall\exists\forall\phi$, where ϕ is an open formula in the language of PV. By Theorem 2 there are polynomial time algorithms $f_1, \ldots, f_r$ where r is a fixed constant such that PV proves the universal closure of the following disjunction with r disjuncts:

$$[f_1(1^{(n)}) = C_n^1 \wedge |C_n^1| \leq cn^k \wedge (|x^1| = n \to C_n^1(x^1) = g(x^1))] \vee$$
$$[f_2(1^{(n)}, x^1) = C_n^2 \wedge |C_n^2| \leq cn^k \wedge (|x^2| = n \to C_n^2(x^2) = g(x^2))] \vee$$
$$\ldots \vee \ [f_r(1^{(n)}, x^1, \ldots, x^{r-1}) = C_n^r \wedge |C_n^r| \leq cn^k \wedge (|x^r| = n \to C_n^r(x^r) = g(x^r))] \ .$$

We shall complete the proof of the theorem by induction on r. The case $r = 1$ and the induction step from $r-1$ to r are analogous, and we describe only the latter. Our induction assumption is that for no polynomial time functions $f'_1, \ldots, f'_{r-1}$ is the disjunction of the form above but with only $r-1$ disjuncts and n large enough provable in PV.

Assume without loss of generality that $k \geq 3$. By Lemma 3 applied to $f \overset{\mathrm{def}}{=} f_1$ and an arbitrary but fixed $\tilde{c}_1 \geq 1$, i.e., using the extra hypothesis

$$\mathrm{PV} \vdash \mathrm{UP}_{k,\tilde{c}_1}(\tilde{f}_1) , \tag{8}$$

there is a constant $c_1 \geq 1$ and a polynomial time algorithm e_1 such that for $n \geq c_1$

$$|e_1(1^{(n)})| = n \wedge C_n^1(e_1(1^{(n)})) \neq g(e_1(1^{(n)})) \ .$$

Substitute $x^1 \overset{\mathrm{def}}{=} e_1(1^{(n)})$ in the disjunction above. That gives for large enough n a valid disjunction of the same form (for different polynomial time functions in place of the f_i's), but with $r-1$ disjuncts:

$$[f_2(1^{(n)}, e_1(1^{(n)})) = C_n^2 \wedge |C_n^2| \leq cn^k \wedge (|x^2| = n \to C_n^2(x^2) = g(x^2))] \vee$$
$$\ldots \vee [f_r(1^{(n)}, e_1(1^{(n)}), x^2, \ldots, x^{r-1}) = C_n^r \wedge |C_n^r| \leq cn^k \wedge (|x^r| = n \to C_n^r(x^r) = g(x^r))] \ .$$

This violates the induction assumption, and completes the induction step.

In the proof we have used the hypotheses that PV proves $\mathrm{UP}_{k,c}(g)$ for some $c \geq 1$, $\mathrm{UP}_{k,\tilde{c}_1}(\tilde{f}_1)$ for some $\tilde{c}_1 \geq 1$, $\mathrm{UP}_{k,\tilde{c}_2}$ for $\tilde{f}_2(1^{(n)}, e_1(1^{(n)}))$, etc., all together $r+1$ such assumptions. Hence one of them must fail. This completes the proof of Theorem 1.

q.e.d.

Making the informal exposition from Section 2 a bit more precise, observe that we do not obtain a hard language directly from a proof of $\mathrm{UP}_{k,c}(g)$. This is done via a iterative process that depends on the provability of additional sentences.

For the reader familiar with the argument in [19, Theorem 1.1], notice that we crucially used that the second application of their initial assumption does not require the uniformity condition. Roughly speaking, this would lead to the consideration of the provability in PV of a sentence expressing a uniform circuit upper bound, while here we are concerned with non-uniform circuit complexity.

Finally, regarding extending Theorem 1 to stronger theories, we remark that in Buss's theory S^1_2 the analogue of Theorem 2 requires a number r of disjuncts that may depend on n, and our induction on parameter r could lead to (composed) functions of super-polynomial complexity.

Acknowledgements. We thank Emil Jeřábek for comments on an initial draft of the paper, which led us to a more robust statement of the result.

References

[1] S. Aaronson and A. Wigderson, Algebrization: A new barrier in complexity theory, *Transactions on Computation Theory*, **1(1)**, (2009).

[2] T. P. Baker, J. Gill and R. Solovay, Relativizatons of the P =? NP question, *SIAM Journal of Computing*, **4(4)**, (1975), pp. 431-442.

[3] S. R. Buss, *Bounded Arithmetic*. Bibliopolis, Naples (1986). (Revision of 1985 Princeton University PhD Thesis.)

[4] 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.

[5] S. A. Cook, Feasibly constructive proofs and the propositional calculus, in: *Proc. 7^{th} Annual ACM Symp. on Theory of Computing*, (1975), pp. 83-97.

[6] S. A. Cook and J. Krajíček, Consequences of the provability of NP $\subseteq$ P/poly, *J. of Symbolic Logic*, **72(4)**, (2007), pp. 1353-1371.

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

[8] E. Jeřábek, Dual weak pigeonhole principle, Boolean complexity, and derandomization, *Annals of Pure and Applied Logic*, **129**, (2004), pp. 1-37.

[9] E. Jeřábek, Approximate counting in bounded arithmetic, *Journal of Symbolic Logic*, **72(3)**, (2007), pp. 959-993.

[10] E. Jeřábek, Approximate counting by hashing in bounded arithmetic, *Journal of Symbolic Logic*, **74(3)**, (2009), pp. 829-860.

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

[12] J. Krajíček, P. Pudlák and G. Takeuti, Bounded arithmetic and the polynomial hierarchy, *Annals of Pure and Applied Logic*, **52**, (1991), pp. 143-153.

[13] J. B. Paris and A. J. Wilkie, On the scheme of induction for bounded arithmetic formulas, *Annals of Pure and Applied Logic*, **35**, (1987), pp. 261-302.

[14] J. Pich, *Complexity Theory in Feasible Mathematics*, PhD Thesis, Charles University in Prague, (2014).

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

[16] J. Pich, Logical strength of complexity theory and a formalization of the PCP theorem in bounded arithmetic, *Logical Methods in Computer Science*, **11(2)**, (2015).

[17] A. A. Razborov, Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, *Izvestiya of the R.A.N.*, **59(1)**, (1995), pp. 201-224.

[18] A. A. Razborov and S. Rudich, Natural proofs, *Journal of Computer and System Sciences*, **55(1)**, (1997), pp. 24-35.

[19] R. Santhanam and R. Williams, On uniformity and circuit lower bounds, *Computational Complexity*, **23**, (2014), pp. 177-205.

Mailing address:

Department of Algebra
Faculty of Mathematics and Physics
Charles University
Sokolovská 83, Prague 8, CZ – 186 75
The Czech Republic