A proof complexity generator

Jan Krajíček*†

Academy of Sciences and Charles University, Prague

Abstract

We define a map $g : \{0,1\}^n \rightarrow \{0,1\}^{n+1}$ such that all output bits are defined by 2DNF formulas in the input bits, and such that g has the following hardness property. For any $b \in \{0,1\}^{n+1} \setminus Rng(g)$, formula $\tau(g)_b$ naturally expressing that $b \notin Rng(g)$ requires exponential size proofs in any proof system for which the pigeonhole principle is exponentially hard.

We define a class of generators generalizing g and show that there is a universal one in this class.

Consider a map $g : x \in \{0,1\}^n \rightarrow y \in \{0,1\}^m$ defined by conditions

$$y_i \equiv \varphi_i(x)$$

where $\varphi_i(x)$ are propositional formulas in $x = (x_1, \ldots, x_n)$ and $m > n$. As the domain of g is smaller than $\{0,1\}^m$ there are $b \in \{0,1\}^m \setminus Rng(g)$. For any such b the formula $\tau(g)_b(x)$:

$$\bigvee_{i \in [m]} b_i \not\equiv \varphi_i(x)$$

expresses that $b \notin Rng(g)$ in the sense that $\tau(g)_b$ is a tautology iff $b \notin Rng(g)$.

Our aim is to define g for which the τ-formulas are hard to prove. When all $\tau(g)_b$ require super-polynomial (resp. exponential) size proofs in a proof system P we say (following [22]) that g **is hard (resp. exponentially hard) proof complexity generator for P**. The τ-formulas have been defined in [7] and independently in [2], and their theory is being developed (see [8, 21, 9, 22, 10, 11, 14]); the introductions to [9] or [22] offer a more comprehensive exposition. The property "$b \notin Rng(g)$" can be expressed by a tautology even for maps g with output bits defined by non-uniform $\mathcal{NP} \cap co\mathcal{NP}$ conditions on the input bits. Such a generality allowed Razborov [22] to formulate an intriguing conjecture about Extended Frege system EF (see also [10]). We do not need such a generality here.

*Keywords: propositional proof complexity, pigeonhole principle.

†Partially supported in part by grants A1019401, AV0Z10190503, MSM0021620839, 201/05/0124, and LC505.

The map g we define is exponentially hard for proof systems for which the pigeonhole principle is exponentially hard. This includes, for example, constant depth Frege systems F_d, polynomial calculus PC or the system from [5] combining the two.

Finally we show that in a class of generators generalizing g, we call them **gadget generators**, there is a universal one.

Exponentially hard generators were previously constructed for resolution R (see [9, Thm.4.2] and [22, Thms.2.10,2.20]). Maps yielding hard τ-formulas for polynomial calculus and a system combing PC with R were constructed in [2] but under the assumption of a particular encoding used in the definition of the τ-formulas (linear encoding, see [2]).

Hard generators are also known to exists (assuming the hardness of factoring) for proof systems admitting feasible interpolation, and our construction applies to systems for which the pigeonhole principle is hard. Note that these two categories of proof systems (not mutually exclusive) cover virtually all[1] proof systems for which a super-polynomial lower bound is known.

The paper is organized as follows. We define the generator in Section 1 and in Section 2 we prove that it is (exponentially) hard for proof systems for which the pigeonhole principle is (exponentially) hard. The class of gadget generators is defined in Section 3 where we construct a universal one in the class.

We assume that the reader has a basic knowledge of proof complexity. In particular, we do not repeat definitions of the proof systems we write about. This background can be found[2] in [6, 18, 15] or in the papers cited at the respective places. However, we do not presume a prior knowledge of proof complexity generators.

Notation: $[n]$ is $\{1, \ldots, n\}$.

1 The generator

Definition 1.1 *Let $k \geq 1$ and put $t := k^2 + k + 1$ and*

$$n := k(k+1) + kt = k^3 + 2k^2 + 2k \ .$$

For an n of this form define map $g_n : \{0,1\}^n \to \{0,1\}^{n+1}$ as follows. Input string x of length n is interpreted as

$$x = (v, u^1, \ldots, u^t)$$

[1] An example of an exception is a constant depth Frege system augmented by PHP as an additional axiom scheme.

[2] This paper accompanies my lecture "Proof complexity and proof search" at the 13th International Congress of Logic, Methodology and Philosophy of Science, Beijing (August 2007). It contains one of the new results mentioned in the talk but not the expository part of the talk. Some of that material can be found in [3, 4, 12, 13].

where

$$v = (v_{ij})_{i\in[k+1], j\in[k]} \quad and \quad u^s = (u^s_j)_{j\in[k]}$$

for $s = 1, \dots, t$. *We call* v *the* gadget[3] variables.

The output string y *of length* $n+1$ *is defined as* $y := (y^1, \dots, y^t)$ *where*

$$y^s_i := \bigvee_{j\in[k]} (v_{ij} \wedge u^s_j)$$

for $s = 1, \dots, t$ *and* $i \in [k+1]$.

Remarks:

(1) We could have defined the generator by conditions

$$y^s_i := \bigwedge_{j\in[k]} (v_{ij} \rightarrow u^s_j) \ .$$

These conditions are equivalent to the original ones assuming that v_{ij}'s satisfy formula $Fn(v)$:

$$\bigwedge_{i\in[k+1]} \bigvee_{j\in[k]} v_{ij} \ \wedge \ \bigwedge_{i\in[k+1]} \bigwedge_{j_1\neq j_2\in[k]} \neg v_{ij_1} \vee \neg v_{ij_2}$$

expressing that $\{(i,j) \mid v_{ij} = 1\}$ is a graph of a function from $[k+1]$ to $[k]$. If we postulate that the output of the map is the zero vector whenever $Fn(v)$ fails, the two definitions would be literally equivalent as, assuming $Fn(v)$, $\neg v_{ij}$ is equivalent to $\bigvee_{j'\neq j} v_{ij'}$.

(2) We could have introduced $\ell \cdot k$ gadget variables v_{ij} with intended meaning to represent maps from $[\ell]$ into $[k]$, for $\ell >> k$ and not just for $\ell = k+1$. The resulting generator would have the output/input ratio about ℓ/k, and its hardness would follow for proof systems where the weak pigeonhole principle PHP^ℓ_k is hard in the same way as Theorem 2.1. However, for proof systems where so weak PHP is hard to prove generators are known already, cf.[19, 22].

(3) In order to treat algebraic proof systems like polynomial calculus PC we can define the generator by degree 2 polynomials. In particular, put

$$y^s_i := \sum_{j\in[k]} v_{ij} \cdot u^s_j \ .$$

Note that these equations define the same map as the original condition assuming $Fn(v)$.

[3] I borrow this term from Razborov's comment.

2 The hardness of generator g

Theorem 2.1 *Let $d \geq 2$. Then for $k = 1, 2, \ldots$ and $n := k^3 + 2k^2 + 2k$ map $g_n : \{0,1\}^n \rightarrow \{0,1\}^{n+1}$ is an exponentially hard proof complexity generator for constant depth Frege systems F_d.*

Proof :

Let $b := (b^1, \ldots, b^t) \in \{0,1\}^{n+1}$ be an arbitrary string, b^s' blocks of length $k+1$. Substitute in an alleged F_d-proof π of $\tau(g_k)_b$ everywhere

$$u_j^s := \bigvee_{i \in [k+1]} v_{ij} \wedge b_i^s \ .$$

The substitution depends on v so we shall denote it $u_j^s(v)$. Denote the substituted proof π'.

Let $\neg PHP_k^{k+1}(v)$ be the formula expressing that v defines a graph of a function violating the pigeonhole principle from $[k+1]$ into $[k]$ (not necessarily bijective):

$$Fn(v) \ \wedge \bigwedge_{i_1 \neq i_2 \in [k+1]} \bigwedge_{j \in [k]} (\neg p_{i_1 j} \vee \neg p_{i_2 j}) \ .$$

Then it is easy to see that there is a size $n^{O(1)} = k^{O(1)}$ F_d-proof σ of

$$\neg PHP_k^{k+1}(v) \rightarrow g_k(v, u^1(v), \ldots, u^t(v)) = b \ .$$

Combining σ and π' gives a proof of $PHP_k(v)$. However, by [1, 16, 17] any such proof must have size exponential in k. Hence π' (and so π too) must have exponential size too.

q.e.d.

Remarks:

(1) We concentrate on F_d as these are the most important proof systems for which no hard generators were previously known. However, the argument utilizes just the hardness of PHP and so it applies to any proof system where PHP is hard and which supports the simple proof manipulations involved (or one of a variety of alternative formalizations of the argument).

(2) In particular, the argument can be modified for polynomial calculus PC. The $\tau(g_n)_b$ formula is represented by the following set of polynomial equations to be refuted:

$$b_i^s = \sum_{j \in [k]} v_{ij} \cdot u_j^s \ . \qquad (1)$$

The measure of complexity of proofs in PC is its degree. Using the dense notation for polynomials, degree d polynomials in n variables are encoded by strings of $O(n^d)$ of field elements. Hence an exponential lower bound on the size corresponds to an $n^{\Omega(1)}$ lower bound on the degree.

The hardness of g_k in PC is derived using the $k/2$ degree lower bound for PC refutations of $\neg PHP_k^{k+1}$ (even with any number $\ell > k$ of pigeons) from [20].

3 Gadget generators

One can consider maps of a general form similar to that of generator g: gadget v is simply a string of $\ell = \ell(k)$ bits and each output block $y^s \in \{0,1\}^{k+1}$ is computed from $u^s \in \{0,1\}^k$ by a fixed polynomial time function f:

$$y^s := f(v, u^s) .$$

In fact, one can take for f the circuit-value function

$$CV_{\ell,k}(v, u)$$

that takes ℓ bits v describing a circuit C with k input bits and $k+1$ output bits and $u \in \{0,1\}^k$, and outputs the string $C(u)$.

For the generator to output more bits than its input has, e.g. $t := \ell + 1$ copies of blocks u^s suffice to swallow the gadget. Thus the only non-canonical part of this construction is the size of the gadget, the parameter ℓ. The observation we want to make is that, in fact, it is possible without a loss of generality to assume that $\ell \leq k^{1+\epsilon}$, any fixed $\epsilon > 0$. This is seen as follows.

Each string $CV_{\ell,k}(v, u^s)$ is computed by a circuit of size $O(\ell + k)$ and the whole generator in size $O(t \cdot (\ell + k))$. Thus for any $\epsilon > 0$ we can take $t > \ell$ large enough but still $t = k^{O(1)}$ such that the generator is computed in time $n^{1+\epsilon}$. Let us call such a generator G.

It is easy to see that the (exponential) hardness of G in a proof system P follows if:

(*) There is any (exponentially) iterable map from $\{0,1\}^k$ to $\{0,1\}^{k+1}$ computed by a circuit of size $\leq \ell(k)^{1/2}$ (in particular, described by $\leq \ell(k)$ bits).

In fact, G is then also (exponentially) iterable. Hence we can now repeat the same construction again, taking blocks u^s of size n and gadgets v of size $n^{1+\epsilon}$. Call this map H.

Hypothesis (*) then implies that there is a size $n^{1+\epsilon}$ gadget describing a circuit computing G for which the corresponding instance of H, and hence H itself, is (exponentially) iterable too.

A general construction like this is unlikely to be useful for lower bounds for specific proof systems. However, a similar universal construction where the gadgets describe the data (a 0-1 matrix and a Boolean function) needed to define the Nisan-Wigderson generator (considered first in this context in [2]) can be helpful.

Acknowledgments:

This paper owns its existence to the encouragement from A. A. Razborov (Princeton) to publish the simple construction.

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] J. Krajíček, A fundamental problem of mathematical logic, *Annals of the Kurt Gödel Society*, Springer-Verlag, Collegium Logicum, Vol.**2**, (1996), pp.56-64.

[4] J. Krajíček, On methods for proving lower bounds in propositional logic, in: *Logic and Scientific Methods* Eds. M. L. Dalla Chiara et al., (Vol. 1 of Proc. of the Tenth International Congress of Logic, Methodology and Philosophy of Science, Florence (August 19-25, 1995)), Synthese Library, Vol.259, Kluwer Academic Publ., Dordrecht, (1997), pp.69-83.

[5] J. Krajíček, Lower bounds for a proof system with an exponential speed-up over constant-depth Frege systems and over polynomial calculus, in: Eds. I.Prívara, P. Růžička, 22nd Inter. Symp. *Mathematical Foundations of Computer Science* (Bratislava, August '97), Lecture Notes in Computer Science 1295, Springer-Verlag, (1997), pp.85-90.

[6] J. Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and Its Applications, Vol. **60**, Cambridge University Press, (1995).

[7] J. Krajíček, On the weak pigeonhole principle, *Fundamenta Mathematicae*, Vol.**170(1-3)**, (2001), pp.123-140.

[8] J. Krajíček, Tautologies from pseudo-random generators, *Bulletin of Symbolic Logic*, **7(2)**, (2001), pp.197-212.

[9] J. Krajíček, Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds, *Journal of Symbolic Logic*, **69(1)**, (2004), pp.265-286.

[10] J. Krajíček, Diagonalization in proof complexity, *Fundamenta Mathematicae*, **182**, (2004), pp.181-192.

[11] J. Krajíček, Structured pigeonhole principle, search problems and hard tautologies, *J. of Symbolic Logic*, **70(2)**, (2005), pp.619-630.

[12] J. Krajíček, Hardness assumptions in the foundations of theoretical computer science, *Archive for Mathematical Logic*, **44(6)**, (2005), pp.667-675.

[13] J. Krajíček, Proof complexity, in: Laptev, A. (ed.), European congress of mathematics (ECM), Stockholm, Sweden, June 27–July 2, 2004. Zurich: European Mathematical Society, (2005), pp.221-231.

[14] J. Krajíček, Substitutions into propositional tautologies, *Information Processing Letters*, **101(4)**, (2007), pp.163-167.

[15] J. Krajíček, Propositional proof complexity I., lecture notes available at `http://www.math.cas.cz/~krajicek/ds1.ps`

[16] J. Krajíček, P. Pudlák,and A. Woods, An Exponential Lower Bound to the Size of Bounded Depth Frege Proofs of the Pigeonhole principle", *Random Structures and Algorithms*, **7(1)**, (1995), pp.15-39.

[17] T. Pitassi, P. Beame, and R. Impagliazzo, Exponential lower bounds for the pigeonhole principle, *Computational complexity*, **3**, (1993), pp.97-308.

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

[19] 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.

[20] A. A. Razborov, Lower bounds for the polynomial calculus, *Computational Complexity*, **7(4)**, (1998), pp.291-324.

[21] A. A. Razborov, Resolution lower bounds for perfect matching principles, in: *Proc. of the 17th IEEE Conf. on Computational Complexity*, (2002), pp.29-38.

[22] A. A. Razborov, Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution, preprint, (May'03).

Mailing address:
Mathematical Institute
Academy of Sciences
Žitná 25, Prague 1, CZ - 115 67
The Czech Republic
`krajicek@math.cas.cz`