Structured pigeonhole principle, search problems and hard tautologies

Jan Krajíček*†

Abstract

We consider exponentially large finite relational structures (with the universe $\{0,1\}^n$) whose basic relations are computed by polynomial size ($n^{O(1)}$) circuits. We study behaviour of such structures when pulled back by $\mathcal{P}/poly$ maps to a bigger or to a smaller universe. In particular, we prove that:

1. If there exists a $\mathcal{P}/poly$ map $g : \{0,1\}^n \to \{0,1\}^m$, $n < m$, iterable for a proof system then a tautology (independent of g) expressing that a particular size n set is dominating in a size 2^n tournament is hard for the proof system.
2. The search problem WPHP, decoding RSA or finding a collision in a hashing function can be reduced to finding a size m homogeneous subgraph in a size 2^{2m} graph.

Further we reduce the proof complexity of a concrete tautology (expressing a Ramsey property of a graph) in strong systems to the complexity of implicit proofs of implicit formulas in weak proof systems.

The weak pigeonhole principle (WPHP) is the statement that no $f : \{0,1\}^m \to \{0,1\}^n$ can be injective if $m > n$. The dual weak pigeonhole principle (dWPHP) is the statement that no $g : \{0,1\}^n \to \{0,1\}^m$ can be surjective if $n < m$. We study the proof complexity of WPHP and dWPHP for $\mathcal{P}/poly$ maps f and g.

---

*The paper was written while the author was a member of the Institute for Advanced Study in Princeton, supported by the NSF grant DMS-0111298.

†On leave from the Mathematical Institute of the Academy of Sciences of the Czech Republic and from the Institute for Theoretical Computer Science of the Charles University, partially supported by grant # A 101 94 01 of the Academy of Sciences and by project LN00A056 of The Ministry of Education of the Czech Republic. A part of this work was done while visiting the Mathematical Institute at Oxford, supported by the EPSRC.

Some information is known. For example, it is a necessary condition for a family of functions to be strongly collision-free that bounded arithmetic theory S^1_2 does not prove WPHP for functions in the family, cf. [11]. Or if RSA were to be secure then WPHP for the modular exponentiation cannot be proved in S^1_2 either, cf.[17]. In these results S^1_2 can be augmented by the true $\forall\Pi^b_1$-theory of $\mathbf{N}$; in particular, by the statements stating the soundness of all propositional proof systems. Consequently we cannot expect to derive hardness results for particular proof systems by appealing to witnessing theorems in bounded arithmetic as such results would automatically apply to all proof systems; we will get some hardness results for particular search problems instead. But we find a link between these search problems and the proof complexity of particular tautologies expressing a Ramsey property of a graph. The main concept used in this link are the implicit proofs of implicit formulas, cf.[14, 15].

For the dWPHP we do not know how to derive proof complexity hardness of dWPHP for some $\mathcal{P}/poly$ map from some established computational complexity conjecture. $\mathcal{P}/poly$ maps for which it is hard to prove (in propositional logic, see Section 1) dWPHP are called proof complexity generators. Maybe the existence of good proof complexity generators is a hypothesis of a different nature than those considered so far in complexity theory. But we will be able to show that the hypothesis implies hardness of some specific tautologies that are independent of any particular generator.

Our method is the "structured WPHP" approach introduced in [11]. In this approach one studies how properties of structures change when the structure is pulled back by a $\mathcal{P}/poly$ map to a bigger or to a smaller universe. As an example of this view a link between resolution complexity of the Ramsey theorem and $R(2)$-complexity[1] of (an instance of) WPHP have been demonstrated in [11], studying structures on $[n]$ given by oraculi. Here we study exponentially large finite relational structures (with the universe $\{0,1\}^n$) whose basic relations are computed by polynomial size ($n^{O(1)}$) circuits.

We do not explicitly use bounded arithmetic (although it is the main source of intuition for us) but occasionally we insert a comment on the bounded arithmetic side of things. I do not recall definitions or facts from bounded arithmetic at these occasions; the reader may find these in [10, 3, 4, 8, 9].

This is a paper in proof complexity and we assume that the reader is

[1] $R(2)$ is a natural extension of resolution operating with clauses formed by literals or conjunctions of two literals, cf.[11].

familiar with its basic concepts, established since [6]. However, we occasionally insert in the text a brief explanatory note. A more recent concept of proof complexity generators is recalled in Section 1.

1 Preliminaries: proof complexity generators

We shall consider propositional proof systems in the sense of [6]. We shall often make an assumption that a proof system P contains resolution R. This is needed for two reasons. The first one is that R proves (by polynomial size proofs) that a computation of a circuit is unique. A circuit is encoded for R in the same way as formulas are, using besides the input variables of the circuit also additional variables (the so called extension variables) for values of subcircuits. The "uniqueness" just means that R proves that the value of any extension variable is uniquely determined by the input variables. Hence R polynomially proves any true boolean sentences $C(a)$ expressed using a circuit C; this phrase means that R proves that the value of the extension atom corresponding to the output of C on input $a \in \{0,1\}^n$ is 1. The second reason for working with proof systems containing R is that that is the assumption needed in Theorem 1.2.

Let g be a $\mathcal{P}/poly$ p-stretching map (cf.[13]). The later assumption means that there is $m = m(n)$, a function of n, such that $m(n) > n$ and $|g(x)| = m(n)$ for all $|x| = n$. Necessarily $m(n) = n^{O(1)}$ if g is $\mathcal{P}/poly$. We will often study g just on inputs from $\{0,1\}^n$ in which case we denote $m(n)$ simply just m.

Let $g : \{0,1\}^n \to \{0,1\}^m$ be a $\mathcal{P}/poly$ p-stretching map. Assume that g restricted to $\{0,1\}^n$ is computed by a circuit C_n, $n \geq 1$. Let $b \in \{0,1\}^m \setminus Rng(g)$. The fact that b is outside of the range of g can be expressed by a size $m^{O(1)}$ tautology denoted $\tau(C_n)_b$; the tautology is just $C_n(x) \neq b$, where x is an n-tuple of boolean variables. Although the tautology depends on C_n and not just on g, the particular circuits C_n often play no role and we occasionally abuse the notation and write just $\tau(g)_b$. Precisely this means that any given statement about $\tau(g)_b$ is claimed for all $\tau(C_n)_b$, for all $\mathcal{P}/poly$ definitions $\{C_n\}_n$ of g.

Proving $\tau(g)_b$, any $b \in \{0,1\}^m$, means proving, in particular, that g is not surjective. Maps g ($\mathcal{P}/poly$ and p-stretching) for which it is hard to prove in a proof system P these τ-formulas are called proof complexity generators for P. There are at least four different level of hardness of g w.r.t. P (cf.[13]) but we shall need here only two, whose definitions we recapitulate now.

For the second part of the following definition we write the τ-formulas as $\tau(g)_b(x)$, showing explicitly the n-tuple of variables x corresponding to the role of x in $g(x) \neq b$.

Definition 1.1 *Let P be a proof system containing R. Let g be a $\mathcal{P}/poly$ p-stretching map.*

1. *Map g is hard for P if for all polynomials $p(m)$, for n large enough no $\tau(g)_b$ for any $b \in \{0,1\}^m \setminus Rng(g)$ has a P-proof of size at most $p(m)$.*

   Map g is exponentially hard for P if there exists $\epsilon > 0$ such that for n large enough no $\tau(g)_b$ for any $b \in \{0,1\}^m \setminus Rng(g)$ has a P-proof of size less than 2^{m^ϵ}.

2. *Map g is called iterable for P if for all polynomials $p(n)$, for all $n \geq 1$ large enough the following holds:*

   Any disjunction of the form

   $$\tau(g)_{B_1}(q^1) \vee \ldots \vee \tau(g)_{B_k}(q^1, \ldots, q^k)$$

   requires a P-proof of size at least $p(n)$. Here $k \geq 1$ is arbitrary, and $B_1, \ldots, B_k$ are circuits with m outputs that are all just substitutions of variables and constants for variables and such that B_1 has no variables, and variables of B_{i+1} are among $q^1, \ldots, q^i$ for $i < k$, where $q^1, \ldots, q^k$ are disjoint m-tuples of variables.

   Map g is called exponentially iterable for P if there exists $\epsilon > 0$ such that the same holds with the lower bound $p(n)$ replaced by 2^{n^ϵ}.

Note that if g is (exponentially) iterable for P then it is also (exponentially) hard for P.

The truth table function **tt** takes as an input a circuit C with k inputs and of size at most $2^{k/2}$ (as encoded by $O(k2^{k/2})$ bits) and outputs the truth table of C, i.e. 2^k bits. Hence **tt** is an example of a $\mathcal{P}/poly$ (canonical circuits based on "circuit-evaluation" compute **tt**) and p-stretching ($n := O(k2^{k/2})$ bits with a fixed O-constant are stretched to $m := 2^k$ bits) map. The following theorem says that it is, in the sense of iterability, the hardest proof complexity generator.

Theorem 1.2 ([13]) *Let P be a proof system containing R. Assume that there is a $\mathcal{P}/poly$ p-stretching map that is (exponentially) iterable for P.*

Then **tt** *is also (exponentially) iterable for P.*

The τ-formulas have been defined in [11] and independently in [1]. The theory of proof complexity generators is being developed, cf. [12, 23, 13, 24]. I shall not describe this development; this can be found in the introductions to [13] or [24].

2 Tournaments

A tournament is a directed graph with exactly one edge between any two distinct vertices: An edge (v, w), directed from v to w, symbolizes that (player) w lost a tournament game to (player) v. A dominating set in a tournament is any set X of its vertices such that any vertex outside X lost a game to some vertex in X.

Assume $m = 2n$. Let T be a tournament with the set of vertices $\{0,1\}^m$. Every such tournament has a dominating set of size m and Erdös [7] has shown that if the directions of the edges are chosen uniformly at random the tournament will, with high probability, have no dominating set of size n. Razborov [21] proved that there are size $m^{O(1)}$ circuits D_m with $2m$ inputs computing the edge relation of a tournament on $\{0,1\}^m$ such that the resulting tournament - which we shall denote T_{m,D_m} - has no dominating set of size n either.

Now let $g : \{0,1\}^n \to \{0,1\}^m$ be a $\mathcal{P}/poly$ p-stretching map computed by a circuit C_n. Define $2n$ input circuit:

$$E_n(x,y) := \begin{cases} D_m(C_n(x), C_n(y)) & \text{if } C_n(x) \neq C_n(y) \\ 1 & \text{if } C_n(x) = C_n(y) \wedge x < y \\ 0 & \text{otherwise} \end{cases}$$

where x, y are n-bit strings ordered lexicographically.

T_{n,E_n} is a tournament and so it has a dominating set $A_n \subseteq \{0,1\}^n$ of size n. This can be expressed by a tautology σ_{n,A_n,C_n,D_m}:

$$\bigvee_{a \in A_n} x = a \vee E_n(a, x)$$

(x is an n-tuples of boolean variables). Now let $B_n := g(A_n)$ be the image of A_n under g in $\{0,1\}^m$. The size of B_n is at most n and so B_n cannot be dominating in T_{m,D_m}. Let $b \in \{0,1\}^m \setminus B_n$ be any vertex not dominated by B_n.

Lemma 2.1 *Assume that σ_{n,A_n,C_n,D_m} has a P-proof of size s. Then $\tau(g)_b$ has a P-proof of size at most $s + n^{O(1)}$.*

Proof :

Reason in P. Start with the size s proof of σ_{n,A_n,C_n,D_m}. Using the definition of E_n, formula σ_n says

$$\bigvee_{a \in A_n} a = x \vee D_m(C_n(a), C_n(x))$$

which implies

$$C_n(x) = b \to [\bigvee_{a \in A_n} a = x \vee D_m(C_n(a), b)] \ .$$

All sentence $D_m(C_n(a), b)$ are false and can be disproved by evaluating them, so we get

$$C_n(x) = b \to [\bigvee_{a \in A_n} C_n(a) = b] \ .$$

But again all $C_n(a) = b$ are false, and so we can derive

$$C_n(x) \neq b \ .$$

That is the formula $\tau(g)_b$. The total size of the proof is s plus $m^{O(1)} = n^{O(1)}$.

q.e.d.

The following theorem is then clear.

Theorem 2.2 *Assume that g is (exponentially) hard for P. Then the tautologies σ_{n,A_n,C_n,D_m} require superpolynomial (resp. exponential) size P-proofs.*

The tautologies σ_{n,A_n,C_n,D_m} do depend on a particular g. Using a stronger hypothesis we get tautologies that are independent of the particular g.

Theorem 2.3 *Assume that P admits (exponentially) iterable $\mathcal{P}/poly$ p-stretching maps. Then the tautologies $\sigma_{n,A_n,\mathbf{tt},D_m}$ require superpolynomial (resp. exponential) size P-proofs.*

Proof :

Assume that P admits (exponentially) iterable $\mathcal{P}/poly$ p-stretching maps. By Theorem 1.2 also the truth-table function **tt** is (exponentially) iterable for P. Hence **tt** is also (exponentially) hard for P and Theorem 2.2 applies.

q.e.d.

3 Vector spaces

Circuits D_m is Section 2 are not canonical and their existence is proved by a probabilistic argument. In this section we use a very canonical structure, the m-dimensional vector space over $\mathbf{F}_2$, but the pull-back of the structure is less elegant as g may not be injective.

We will consider vector spaces over $\mathbf{F}_2$ in the following language: ternary relation $R(x, y, z)$ standing for the graph of the addition on the space, and binary relation $S(x, y)$ computing (by its truth value) the scalar product. The axioms of *partial* vector spaces are the usual axioms about addition and scalar product in vector spaces rewritten using R and S. We do not include the axiom $\forall x, y \exists z, R(x, y, z)$ that would say that the addition as given by R is a total function. Note that all axioms of partial vector spaces are thus universal sentences. A structure $W = (X, R, S)$ in this language with universe X is a *partial* vector space over $\mathbf{F}_2$ iff it satisfies all these universal axioms.

Let $\oplus_m$ and $\langle\ ,\ \rangle$ be the (coordinate-wise) addition and the scalar product on the canonical vector space V_m on $\{0,1\}^m$, with $\overline{0}$ the zero vector and $(0, \ldots, 0, 1, 0, \ldots, 0)$'s the basis vectors.

Now let $g : \{0,1\}^n \to \{0,1\}^m$ be a $\mathcal{P}/poly$ p-stretching map computed by a circuit C_n. Define a structure $W' = (\{0,1\}^n, R'_n, S'_n)$ by:

$$R'_n(x, y, z) \ \text{ iff } \ C_n(x) \oplus_m C_n(y) = C_n(z)$$

and

$$S'_n(x, y) \ \text{ iff } \ \langle x, y\rangle = 1 \ .$$

Structure W' is not necessarily a (partial) vector space because one point of V_m could have been pulled-back to several points in $\{0,1\}^n$, as g may not be injective. This we remedy by taking a quotient of W' modulo the equivalence relation:

$$x \sim y \ \text{ iff } \ C_n(x) = C_n(y) \ .$$

Define R_n and S_n to be the quotients of R'_n and S'_n by $\sim$ respectively, and put $W := (\{0,1\}^n/\sim, R_n, S_n)$.

If W were a total vector space then there would exist $u_1, \ldots, u_n \in \{0,1\}^n$ such that no vector in W could be orthogonal to all $u_i/\sim$. If it were only a partial vector space then there would be $u_1, u_2 \in \{0,1\}^n$ such that $R_n(u_1/\sim, u_2/\sim, c/\sim)$ could hold for no $c \in \{0,1\}^n$.

Lemma 3.1 *There is a sequence* $U_n = (u_1, \ldots, u_n)$ *of* n *elements of* $\{0,1\}^n$ *satisfying the following tautology* ρ_{n,U_n,C_n}*:*

$$\neg R_n(u_1/\sim, u_2/\sim, y/\sim) \ \vee \ \bigvee_{u \in U_n} S_n(u/\sim, x/\sim) \ .$$

x *and* y *being* n*-tuples of boolean variables.*

Theorem 3.2 *Assume* ρ_{n,U_n,C_n} *has a* P*-proof of size* s*. Then there are* $b, c \in \{0,1\}^m$ *such that the disjunction* $\tau(g)_b \vee \tau(g)_c$ *has a* P*-proof of size* $s + n^{O(1)}$*.*

In particular, if g *is (exponentially) iterable for* P *then the formulas* ρ_{n,U_n,C_n} *require superpolynomial (resp. exponential) size* P*-proofs.*

Proof :

Let $c := g(u_1) \oplus_m g(u_2)$ and let $b \in \{0,1\}^m$ be an element orthogonal to all $g(u)$, $u \in U_n$. It exists as the dimension of V_m is $m > n$.

Reason in P, starting with a proof of ρ_{n,U_n,C_n}. Formula ρ_{n,U_n,C_n} means, by the definitions of R_n and S_n:

$$g(u_1) \oplus_m g(u_2) \neq g(y) \ \vee \ \bigvee_{u \in U_n} \langle g(u), g(x) \rangle = 1 \ .$$

As $g(u_1) \oplus_m g(u_2) = c$ and $\bigvee_{u \in U_n} \langle g(u), b \rangle = 0$ are true boolean sentences, P deduces:

$$g(x) \neq b \ \vee \ g(y) \neq c$$

which is just the formula $\tau(g)_b \vee \tau(g)_c$.

The size of the whole proof is $s + n^{O(1)}$.

q.e.d.

Analogously to Section 2 we can replace g by the canonical truth-table function.

Theorem 3.3 *Assume that* P *admits (exponentially) iterable* $\mathcal{P}/poly$ *maps. Then the tautologies* $\rho_{n,U_n,\mathbf{tt}}$ *require superpolynomial (resp. exponential) size* P*-proofs.*

4 Homogeneous subgraphs

We have studied maps for which it is hard to prove dWPHP, i.e. which are good proof complexity generators, getting hardness of some tautologies as a result. The disadvantage of that is that the existence of good proof complexity generators has not been proved so far from any of the usual complexity theoretic assumptions. In this section we will look at maps for which the ordinary WPHP is hard to prove. Such maps are known to exists under plausible assumptions. We get the hardness of search problems as a direct result, not the hardness of tautologies. However, there is another less direct link to hardness of tautologies; this will be in Section 6.

We shall consider undirected graphs with vertices $\{0,1\}^n$ where the edge relation is computed by circuits C with $2n$ inputs. Such graphs will be denoted $G_{n,C}$. By Ramsey theorem every such graph contains a homogeneous set (a clique or an independent set) of size at least $n/2$. On the other hand, as shown by Erdös [7], random graph has no homogeneous set of size $2n$. Razborov [21] has shown that there is a circuit R_n with $2n$ inputs and of size $n^{O(1)}$ such that the graph G_{n,R_n} also does not have a homogeneous set of size $2n$.

We shall define two particular search problems and later we discuss a many-one reducibility among them, without defining a general notion of a search problem. This does not seem to leave a room for a confusion but the reader may find general definitions in [3, 2, 8, 9].

Definition 4.1 *1. The search problem RAM asks for the following. Given a pair $1^{(m)}$ (a canonical string of length m) and a $4m$-input circuit D find a homogeneous subgraph (by listing its vertices) of size m in $G_{2m,D}$.*

2. The search problem WPHP asks for the following. Given a triple $1^{(n)}$, $1^{(m)}$ and a circuit E with m inputs and n outputs such that $n < m$, find a pair $u \neq v$ of distinct elements of $\{0,1\}^m$ such that $E(u) = E(v)$.

Theorem 4.2 *The search problem WPHP can be $\mathcal{P}/poly$ many-one reduced to RAM.*

Proof :

Let $n < m$ and E be an input to WPHP. By amplifying the map defined by E if necessary we may assume that $4n \leq m$. This amplification is quite standard and goes back to [19]. For example, if $m = n+1$ then a map defined by $E(E(x_1, \ldots, x_{n+1}), x_{n+2})$ maps $\{0,1\}^{n+2}$ into $\{0,1\}^n$ and a collision in this map yields a collision in E (in other words, if E violates WPHP so does the amplified map). By iterating this procedure we can boost $m = n+1$ to $m = 4n$ in polynomially many steps.

Define a $2m$-input circuit D by:

$$D(u, v) \ := \ R_n(E(u), E(v))$$

where u and v are m-tuples of boolean variables.

Consider the graph $G_{m,D}$ as an input to RAM and let H be a homogeneous subgraph of $G_{m,D}$ of size $m/2 \geq 2n$. As G_{n,R_n} has no that large homogeneous subgraphs the map E cannot be injective on H and the wanted collision of E can be found among the elements of H (by exhaustive search in polynomial time).

q.e.d.

Note that the non-uniform part of the reduction is only in the choice of circuits R_n.

A family $h_y(x)$ of p-time functions from $\{0,1\}^{\ell(|y|)}$ into $\{0,1\}^{\ell(|y|)-1}$, where $\ell(n)$ is a polynomial, is a *strongly collision-free family of hash functions* if there is no polynomial-time function f that on y computes $x_1 < x_2 \in \{0,1\}^{\ell(|y|)}$ with $h_y(x_1) = h_y(x_2)$ (cf. [25]). An example of a family of functions with this property (unless the discrete logarithm is tractable) is the Cham - van Heijst - Pfitzman family, see [25, Chpt.7].

The following is an immediate corollary of Theorem 4.2.

Corollary 4.3 *Finding collisions in a family $h_y(x)$ of p-time functions as above can be $\mathcal{P}/poly$ many-one reduced to RAM.*

Corollary 4.4 *The task to decode RSA can be $\mathcal{P}/poly$ many-one reduced to RAM.*

Proof :

Let $f_n(a, b, x)$ be the function:

$$f_n(a, b, x) \ := \ a^x \ (\mathrm{mod}\ b)$$

where a and b are two parameters of length $n \geq 1$ and x is arbitrary.

Put $m := 4n$. Let $E_{n,m}(a, b, x)$ be some fixed canonical circuit computing $f_n(a, b, x)$ on inputs of length m, for arbitrary a and b.

By Theorem 4.2 we can find, employing RAM, two distinct elements $u, v \in \{0,1\}^m$ for which $f_n(a, b, u) = f_n(a, b, v)$. Hence we also get a non-zero $w := u - v$ such that $a^w \equiv 1 (\mathrm{mod}\ b)$.

It is known that having such w is enough to break the RSA (see [17, Thm.3] for a similar argument).

q.e.d.

It is known that in the oracle setting (circuit D is replaced by an oracle) RAM is not even Turing reducible to PLS (polynomial local search), cf.[4]. It would be interesting (because of the bounded arithmetic consequences, cf.[3, 4, 8, 9]) to show that RAM cannot be reduced to GLS (generalized local search, cf.[3]) or at least to MIN (finding a minimal element in a partial ordering, cf.[3, 8, 9]).

We leave it to the reader to investigate the pull-back of the canonical vector space V_n on $\{0,1\}^n$ by f; the results stated above for RAM can be analogously proved for the following search problem: Given string $1^{(m)}$ and circuits computing relations $R_m(x,y,z)$ and $S_m(x,y)$ on $\{0,1\}^m$ defining a partial vector space find m distinct points in $\{0,1\}^m$ all orthogonal (as computed by S_m) to each other.

5 Intermezzo: implicit proofs

We need to recall the definition of implicit proofs before the next section.

By [6] a proof system is a polynomial-time function Q whose range is exactly the set $TAUT$ of tautologies in the DeMorgan language. A Q-proof of a formula τ is any string π such that $Q(\pi) = \tau$. The idea of implicit proofs from [14] is that instead of representing π of length ℓ by writing down all it's ℓ bits π_i we present a circuit β with $\log(\ell)$ inputs that computes π_i from $i \leq \ell$. This implicit description of π may be, in principle, exponentially smaller than π. The circuit β alone does not constitute a proof of anything and in order to get a proof system in the sense of [6] we supplement β with an ordinary P-proof α of the fact that β indeed describes a valid Q-proof.

Let us describe this a some detail. Assume that the computations of Q are done by a deterministic machine (also denoted Q) running in time n^c. We will represent the computation an input of size n by the list of all $t \leq n^c$ instantaneous descriptions of the computation. This list can be represented by an $t \times O(t)$ 0-1 matrix W: think of the ith row W_i as representing the ith instantaneous description. We may assume that t is a power of 2 and that W is a $t \times t$ matrix (by increasing t to $O(t)$ if needed). Let $k := \log(t)$ and let $\beta(i,j)$, $i = (i_1, \ldots, i_k)$ and $i = (j_1, \ldots, j_k)$, be a circuit with $2k$ inputs.

Propositional formula $Correct_\beta^Q$ is the canonical propositional formula expressing that:

- The matrix $W_{i,j} := \beta(i,j)$ satisfies all local conditions in order to be a valid computation of Q on an input (encoded in the first row W_1).

The size of $Correct_\beta^Q$ is $O(|\beta|)$.

Definition 5.1 *Let P, Q be any proof systems and assume that P contains R. Proof system $[P,Q]$ is defined as follows: A $[P,Q]$-proof of $\tau \in TAUT$ is a pair (α, β) such that:*

1. *β is a single-output boolean circuit in variables $(i_1, \ldots, i_k, j_1, \ldots, j_k)$, some $k \geq 1$.*
2. *β defines a valid computation W of Q (on some input) whose output is τ.*
3. *α is a P-proof of the tautology $Correct_\beta^Q$.*

Note that we need not to ask for a P-proof of the fact that the output of W is τ as that is a true boolean sentence.

In defining $[P,Q]$ we have restricted to proofs of ordinary (explicitly given) formulas τ. But in fact, we could have defined proofs of formulas themselves given implicitly by a circuit; cf.[15]. We will not give a general definition here but only two particular cases that we need in Section 6.

Let us fix a useful notation. Let n, m, s be three parameters. Let x be an m-tuple of variables and w an s-tuple of variables. $Cir_{n,m,s}(x,w)$ is a circuit that interprets w as a code of a circuit C with m inputs and n outputs, and computes the value of C on x. Hence $Cir_{n,m,s}$ has n outputs. Note that the size of $Cir_{n,m,s}$ is $O(s)$ if $n, m \leq s$.

Definition 5.2

1. *Let $1 \leq n < m \leq s$. Let i, j be m-tuples of variables and w an s-tuple of variables. Circuit*

   $$\gamma_{n,m,s}^{\neg WPHP}(i,j,w)$$

   has $2m+s$ inputs. On an input i, j, w the circuit outputs:

   - *Formula $Cir_{n,m,s}(i,w) \neq Cir_{n,m,s}(j,w)$, if $i \neq j$.*
   - *Formula 1, if $i = j$.*

2. *Let $1 < m \leq s$. Let $i^1, \ldots, i^m$ be $2m$-tuples of disjoint variables, j a single variable, and let w be an s-tuple of variables. Circuit:*

   $$\gamma_{m,s}^{\neg RAM}(i^1, \ldots, i^m, j, w)$$

   is a circuit with $2m^2 + s + 1$ inputs that outputs:

- *Formula* $\bigvee_{u<v\leq m} Cir_{1,4m,s}(i^u, i^v, w)$*, if* $j = 1$ *and all* $i^1, \dots, i^m$ *are distinct* $2m$*-tuples.*
- *Formula* $\bigvee_{u<v\leq m} \neg Cir_{1,4m,s}(i^u, i^v, w)$*, if* $j = 0$ *and all* $i^1, \dots, i^m$ *are distinct* $2m$*-tuples.*
- *Formula* 1*, otherwise.*

The size of $\gamma^{\neg WPHP}_{n,m,s}$ is $O(s)$ and the size of $\gamma^{\neg RAM}_{m,s}$ is $O(m^2 s)$, if $n \leq m \leq s$.

Note the different role of variables in the γ-formulas: variables i and j are used to enumerate clauses of the implicitly defined formula (set of clauses), while variables w are free parameters.

We will use the following lemma in Section 6. Recall that R^* is the tree-like resolution.

Lemma 5.3 ([14, L.4.1]) *If* P *contains* R *then* $[P, R^*]$ *p-simulates* P*.*

Now we state a theorem that will be used only for an illustration in Section 6, and so we only sketch its proof (using bounded arithmetic). Recall that EF is the Extended Frege proof system of [6].

Theorem 5.4 *Assume* $1 \leq n < m \leq s$*. Both formulas* $\gamma^{\neg WPHP}_{n,m,s}$ *and* $\gamma^{\neg RAM}_{m,s}$ *have size* $s^{O(1)}$ $[EF, EF]$*-refutations.*

Proof sketch :

It is shown in [14, Thm.2.1] that $[EF, EF]$ simulates bounded arithmetic theory V^1_2. The simulation is done by a witnessing argument and applies to simulations of proofs of sequents of $\Sigma^{1,b}_1$-formulas (this is what is done in the proof of [14, Thm.2.1] although the theorem is stated only for Π^b_1-consequences of V^1_2). For both WPHP and RAM the statements that these search problems are defined on all inputs are expressed even by Σ^b_1-formulas and propositional translations of (negations of) these formulas are the implicit formulas from Definition 5.2.

The theorem follows as it is easy to see that V^1_2 proves that both WPHP and RAM are defined everywhere.

q.e.d.

Simulations of bounded arithmetic theories using implicit proof systems can be proved for much weaker theories than V^1_2 (in [14] we were interested in strong theories). In particular, Theorem 5.4 holds with $[P, Q]$ in place of $[EF, EF]$ where both P and Q are much weaker than EF.

6 Ramsey graphs and hard tautologies

We will now reexamine RAM using the implicit formulas and proofs. Circuits R_n are those from Section 4. The size of the circuits, as computed in [21], is $O(n^5 \log n) < n^6$.

Definition 6.1 *Let $x^1, \ldots, x^{2n}$ be n-tuples of distinct variables. The formula ω_{n,R_n} is the formula:*

$$\bigwedge_{i \neq j} x^i \neq x^j \;\rightarrow\; [\bigvee_{i \neq j} R_n(x^i, x^j) \;\vee\; \bigvee_{i \neq j} \neg R_n(x^i, x^j)]$$

where i, j range over $\{1, \ldots, 2n\}$.

The formula expresses that G_{n,R_n} has no homogeneous subgraph of size $2n$ and it is a tautology.

Formulas $\gamma^{\neg WPHP}$ and $\gamma^{\neg RAM}$ are sets of clauses formed by formulas not by literals, but in the following theorem we shall speak about $[P, Q]$-refutations of these formulas and Q could be even R^*. In such a case we tacitly assume (but not include explicitly in the notation as that would make the formulas quite unreadable) that the formulas in the clauses are reduced to literals using limited extension, as it is customary in resolution, cf.[6] (or the first paragraph in Section 1). Alternatively one could think that the R^*-refutations can also operate (via some complete set of sound schematic inference rules, the so called Frege rules in the terminology of [6]) directly with formulas but only with those which appear as subformulas of some formulas in one of the original clauses of $\gamma^{\neg WPHP}$ or $\gamma^{\neg RAM}$ respectively.

Theorem 6.2 *Let P, Q be two proof systems, P containing R and Q containing R^*. Assume that:*

- *For any function $n \leq s(n) \leq n^{O(1)}$ there is a $[P, Q]$-refutation of $\gamma^{\neg RAM}_{4n,s(n)}$ of size $n^{O(1)}$.*
- *There is a function $n \leq t(n) \leq n^{O(1)}$ such that any $[P, Q]$-refutation of $\gamma^{\neg WPHP}_{n,4n,t(n)}$ must have the size $n^{\omega(1)}$, i.e. superpolynomial in n.*

Then the formulas ω_{n,R_n} require superpolynomial (size $n^{\omega(1)}$) P-proofs.

Proof :

Let us take $t = t(n)$ satisfying the second hypothesis, and let us fix $s = O(n^6 + t)$ such that any circuit of the form $R_n(E(x), E(y))$ is encoded by s bits if E is encoded by t bits. So s is also $n^{O(1)}$.

To simplify the notation let us denote the circuit encoded by a t-tuple w in $\gamma^{\neg WPHP}_{n,4n,t}$ simply by E_w instead of $Cir_{n,4n,t}(x, w)$.

Define a circuit D_w with $8n$ inputs to be $R_n(E_w(x), E_w(y))$, and let w' be the $\leq s$ bits encoding it. The code w' can be computed by a circuit from w, say by $S(w)$.

Now we use the first hypothesis and substitute everywhere in the $[P, Q]$-refutation of $\gamma^{\neg RAM}_{4n,s}$ code $S(w)$ in place of the s variables used in $\gamma^{\neg RAM}_{4n,s}$ for the circuit encoding.

After this substitution the clauses of $\gamma^{\neg RAM}_{4n,s}$ become:

$$[\bigvee_{i \neq j \in H} R_n(E_w(i), E_w(j)) \ \vee \ \bigvee_{i \neq j \in H} \neg R_n(E_w(i), E_w(j))]$$

where H ranges over all sets of $2n$ distinct vertices of $\{0, 1\}^{4n}$. Call this disjunction δ_H.

Let π be a P-proof of ω_{n,R_n} of size ℓ. We may substitute the $2n$ different tuples of variables x^u by the $2n$ tuples $E_w(i)$, for $i \in H$. This would get us a P-proof π_H of

$$\bigwedge_{i \neq j \in H} E_w(i) \neq E_w(j) \ \rightarrow \ \delta_H \ .$$

The size of π_H is $O(\ell |E_w|) = O(\ell t) = \ell n^{O(1)}$.

Now we would like to use proof π_H and to derive all clauses δ_H of $\gamma^{\neg RAM}_{4n,s}$ from formulas $\bigwedge_{i \neq j \in H} E_w(i) \neq E_w(j)$, and then continue in the refutation as in the $[P, Q]$-refutation of $\gamma^{\neg RAM}_{4n,s}$. However, that would combine P-proofs with Q-proofs and we would not get a $[P, Q]$-refutation of $\gamma^{\neg WPHP}_{n,4n,s}$ as we want. But we can proceed indirectly.

By Lemma 5.3 P is p-simulated by $[P, R^*]$. Use this simulation to get from P-proofs π_H $[P, R^*]$-proofs π'_H (of the same formula) of size $(\ell n)^{O(1)}$, and then proceed as described above to get a $[P, Q]$-refutation (here we use that Q contains R^*) of formulas $\bigwedge_{i \neq j \in H} E_w(i) \neq E_w(j)$, one for each H. However, it is easy to see that each of these formulas has a polynomial size R^*-derivation from clauses of $\gamma^{\neg WPHP}_{n,4n,t}$, and quite uniformly described in terms of H, and hence we get a $[P, Q]$-refutation of $\gamma^{\neg WPHP}_{n,4n,s}$.

The total size of this refutation is polynomial in ℓ and n (as both $t(n)$ and $s(n)$ are), and so the theorem follows.

q.e.d.

Judging from what is known about WPHP and RAM in bounded arithmetic, it is consistent with the present knowledge that the two hypotheses in Theorem 6.2 are fulfilled by some $[P, Q]$, where Q is R or one of $R(k)$ of [11], with $1 \leq k < O(\log 2^{O(n)}) = O(n)$. Having weak P would not be necessarily bad as $[P, Q]$ is p-equivalent to $[[P, Q], Q]$ for many natural P, Q (see [14, L.4.2] and the remark thereafter; this can be extended to proofs of implicit formulas) and $[P, Q]$ can be much stronger than P. These remarks will be expanded upon elsewhere.

References

[1] 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.

[2] P. Beame, S. A. Cook, J. Edmonds, R. Impagliazzo, and T. Pitassi, The Relative Complexity of NP Search Problems, in: *Proc. 27th Annual ACM Symposium on the Theory of Computing*, (1995), pp. 303-314.

[3] M. Chiari and J. Krajíček, Witnessing functions in bounded arithmetic and search problems, *J. of Symbolic Logic*, **63(3)**, (1998), pp. 1095-1115.

[4] M. Chiari and J. Krajíček, Lifting independence results in bounded arithmetic, *Archive for Mathematical Logic*, **38(2)**, (1999), pp.123-138.

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

[6] S. A. Cook and A. R. Reckhow, The relative efficiency of propositional proof systems, *J. Symbolic Logic*,**44(1)**, (1979), pp.36-50.

[7] Erdös, P., Some remarks on the theory of graphs, *Bull. of the AMS*, **53**, (1947), pp.292-294.

[8] J. Hanika, *Search problems and bounded arithmetic*, PhD Thesis, Charles University, (2004).

[9] J. Hanika, Herbrandizing search problems in bounded arithmetic, *Mathematical Logic Quarterly*, **50(6)**, pp.577-586, (2004).

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

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

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

[13] J. Krajíček, Dual weak pigeonhole principle, pseudo-surjective functions, and provability of circuit lower bounds, *Journal of Symbolic Logic*, to app. (preprint Nov.'02).

[14] J. Krajíček, Implicit proofs, *J. of Symbolic Logic*, **69(2)**, pp.387-397, (2004).

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

[16] J. Krajíček, and P. Pudlák, Propositional proof systems, the consistency of first order theories and the complexity of computations, *J. Symbolic Logic*, **54(3)**, (1989), pp.1063-1079

[17] J. Krajíček, P. Pudlák, Some consequences of cryptographical conjectures for S^1_2 and EF, *Information and Computation*, Vol. **140 (1)**, (January 10, 1998), pp.82-94.

[18] A. Maciel, T. Pitassi, and A. Woods, A new proof of the weak pigeonhole principle. Maciel, *J. of Computer and Systems Science*, **64**, (2002), pp.843-872.

[19] J. B. Paris, A. J. Wilkie, and A. Woods, Provability of the pigeonhole principle and the existence of infinitely many primes. *Journal of Symbolic Logic*, **53**, (1988), pp.1235-1244.

[20] R. Raz, Resolution Lower Bounds for the Weak Pigeonhole Principle, in: *Proc. of the 34th STOC*, (2002), pp.553-562.

[21] A. A. Razborov, Formulas of bounded depth in the basis (&,$\oplus$) and some combinatorial problems. (Russian) *Vopr. Kibern.*, Moscow, Vol.**134**, (1988), pp.149-166.

[22] A. A. Razborov, Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, *Izv. Ross. Akad. Nauk Ser. Mat.*, **59(1)**, (1995), pp. 201-224.

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

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

[25] D. R. Stinson, *Cryptography: theory and practice*, CRC Press LLC, (1995).

Mailing address:

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