A reduction of proof complexity to computational complexity for $AC^0[p]$ Frege systems

Jan Krajíček

Faculty of Mathematics and Physics
Charles University in Prague

Abstract

We give a general reduction of lengths-of-proofs lower bounds for constant depth Frege systems in DeMorgan language augmented by a connective counting modulo a prime p (the so called $AC^0[p]$ Frege systems) to computational complexity lower bounds for search tasks involving search trees branching upon values of maps on the vector space of low degree polynomials over $\mathbf{F}_p$.

In 1988 Ajtai [2] proved that the unsatisfiable set ($\neg\mathrm{PHP}_n$) of propositional formulas

$$\bigvee_{j\in[n]} p_{ij} \quad \text{and} \quad \neg p_{i_1 j} \vee \neg p_{i_2 j} \quad \text{and} \quad \neg p_{i j_1} \vee \neg p_{i j_2}$$

for all $i \in [n+1] = \{1, \dots, n+1\}$, all $i_1 \neq i_2 \in [n+1], j \in [n]$, and all $i \in [n+1], j_1 \neq j_2 \in [n]$ respectively, expressing the failure of the pigeonhole principle (PHP), has for no $d \geq 1$ a polynomial size refutation in a Frege proof system operating only with DeMorgan formulas of depth at most d. Subsequently Krajíček [18] established an exponential lower bound for these so called AC^0 Frege proof systems (for different formulas) and Krajíček, Pudlák and Woods [23] and Pitassi, Beame and Impagliazzo [26] improved independently (and announced jointly in [7]) Ajtai's bound for PHP to exponential.

All these papers employ some adaptation of the random restriction method that has been so successfully applied earlier in circuit complexity (cf. [1, 14, 31, 15]). Razborov [28] invented already in 1987 an elegant method, simplified and generalized by Smolensky [30], to prove lower bounds even for $AC^0[p]$ circuits, p a prime. Thus immediately after the lower bounds for AC^0 Frege systems were proved researchers attempted to adapt the Razborov-Smolensky method to proof complexity and to prove lower bounds also for $AC^0[p]$ Frege systems.

This turned out to be rather elusive and no lower bounds for the systems were proved, although some related results were obtained. Ajtai [3, 4, 5], Beame et.al.[6] and Buss et.al.[9] proved lower bounds for AC^0 Frege systems in DeMorgan language augmented by the so called modular counting principles as extra axioms (via degree lower bounds for the Nullstellensatz proof system in [6, 9]), Razborov [29] proved $n/2$ degree lower bound for refutations of ($\neg\mathrm{PHP}_n$) in polynomial calculus PC of Clegg, Edmonds and Impagliazzo [12], and Krajíček [21] used methods of Ajtai [4, 5] to prove $\Omega(\log\log n)$ degree lower bound for PC proofs of the counting principles. Krajíček [20] proved an exponential lower bound for a subsystem of an $AC^0[p]$ Frege system that extends both constant depth Frege systems and polynomial calculus, Maciel and Pitassi [24] demonstrated a quasi-polynomial simulation of $AC^0[p]$ proof systems by a proof system operating with depth 3 threshold formula, Impagliazzo and Segerlind [17] proved that AC^0 Frege systems with counting axioms modulo a prime p do not polynomially simulate polynomial calculus over $\mathbf{F}_p$, and recently Buss, Kolodziejczyk and Zdanowski [10] proved that an $AC^0[p]$ Frege system of any fixed depth can be quasi-polynomially simulated by the depth 3 $AC^0[p]$ system. Also, Buss et.al.[9] showed that the $AC^0[p]$ Frege systems are polynomially equivalent to the Nullstellensatz proof system of Beame et.al.[6] augmented by the so called extension axioms formalizing in a sense the Razborov-Smolensky method.

In this paper we reduce the task to prove a lengths-of-proofs lower bound for $AC^0[p]$ Frege systems to the task to establish a computational hardness of a specific computational task. The task is a search task and it is solved by trees branching upon values of maps on the vector space of low degree polynomials over $\mathbf{F}_p$. The hardness statements to which lower bounds are reduced say that every tree of small depth and using small degree polynomials succeeds only with an exponentially small probability.

Maciel and Pitassi [25] formulated such a reduction of proof complexity to computational complexity (and the implied conditional lower bound). However, in their construction they needed to redefine the proof systems (allowing arbitrary formulas with $MOD_{p,i}$ connectives and restricting only cut-formulas to constant depth) and the hard examples whose short proofs yield a computational information are not $AC^0[p]$ formulas. In particular, their reduction does not seem to yield anything for the originally defined $AC^0[p]$ Frege systems (see Section 1).

The paper is organized as follows. In Section 1 we recall the definition of the proof systems. In Sections 2 - 5 we reduce the lower bounds to the task to show the existence of winning strategies for a certain game. This is reduced further in Section 6 to the task to show that search trees of small depth that branch upon values of maps on the vector space of low degree polynomials over $\mathbf{F}_p$ cannot solve a certain specific computational task.

More background on proof complexity can be found in [19] or [27], the problem (and a relevant background) to prove the lower bound for the systems is discussed in detail also in [22, Chpt.22].

1 $AC^0[p]$ Frege proof systems

We will work with a sequent calculus style proof system in a language with connectives $\neg$, unbounded arity $\bigvee$ and unbounded arity connectives $MOD_{p,i}$

for p a prime and $i = 0, \ldots, p-1$. The intended meaning of the formula $MOD_{p,i}(y_1, \ldots, y_k)$ is that $\sum_i y_i \equiv i \pmod p$. The proof system has the usual structural rules (weakening, contraction and exchange), the cut rule, the left and the right $\neg$ introduction rules and two introduction rules for $\bigvee$ modified for the unbounded arity; the $\bigvee : left$ rule

$$\frac{\varphi_1, \Gamma \to \Delta \quad \varphi_2, \Gamma \to \Delta \quad \ldots \quad \varphi_t, \Gamma \to \Delta}{\bigvee_{i \le t} \varphi_i, \Gamma \to \Delta}$$

and the $\bigvee : right$ rule

$$\frac{\Gamma \to \Delta, \varphi_j}{\Gamma \to \Delta, \bigvee_{i \le t} \varphi_i}$$

any $j \le t$. There are no rules concerning the $MOD_{p,i}$ connectives but there are new **MOD$_p$-axioms** (we follow [19, Sec.12.6]):

- $MOD_{p,0}(\emptyset)$
- $\neg MOD_{p,i}(\emptyset)$, for $i = 1, \ldots, p-1$
- $MOD_{p,i}(\Gamma, \phi) \ \equiv \ [(MOD_{p,i}(\Gamma) \wedge \neg\phi) \vee (MOD_{p,i-1}(\Gamma) \wedge \phi)]$
  for $i = 0, \ldots, p-1$, where $i-1$ means $i-1$ modulo p, and where Γ stands for a sequence (possibly empty) of formulas.

The depth of the formula is the maximal number of alternations of the connectives; in particular, formulas from ($\neg$PHP$_n$) have depth 1 and 2 respectively. We have not included among the connectives the conjunction $\bigwedge$; this is in order to decrease the number of cases one needs to consider in the constructions later on. Note that the need to express $\bigwedge$ using $\neg$ and $\bigvee$ may increase the depth of AC^0 formulas comparing to how it is usually counted. But as we are aiming at lower bounds for all depths this is irrelevant.

We shall denote the proof system $LK(MOD_p)$ and its depth d subsystem (operating only with formulas of depth at most d) $LK_d(MOD_p)$. It is well-known that this system is polynomially equivalent to constant depth Frege systems with $MOD_{p,i}$ connectives (or to Tait style system as in [10]) and in the mutual simulation the depth increases only by a constant as the systems have the same language (cf.[19]). The size of a formula or of a proof is the total number of symbols in it.

2 From a proof to a game with formulas

In this section and in the next one we define certain games using the specific case of the PHP as an example. This is in order no to burden the presentation right at the beginning with a technical discussion of the form of formulas we allow. As it is shown in Section 6 this is without a loss of generality and, in fact, motivates the general formulation there.

Consider the following game $G(d, n, t)$ played between two players, Prover and Liar. At every round Prover asks a question which Liar must answer. Allowed questions are:

(P1) What is the truth-value of φ?

(P2) If Liar already gave a truth-value to $\varphi = \bigvee_{i \leq u} \varphi_i$, Prover can ask as follows:

   (a) If Liar answered false then Prover can ask an extra question about the truth value of any one of φ_j, $j \leq u$.

   (b) If Liar answered true then Prover can request that Liar witnesses his answer by giving a $j \leq u$ and stating that φ_j is true.

All formulas asked by Prover are built from the variables of ($\neg$PHP$_n$), and must have the depth at most d and the size at most 2^t. The Liar's answers must obey the following rules:

(L0) When asked about a formula he already gave a truth value to in an earlier round Liar must give the same answer.

(L1) He must give φ and $\neg\varphi$ opposite truth values.

(L2) If asked according to (P2a) about φ_j he must give value false. If asked according to (P2b) he must give value true also to some φ_j with $j \leq u$.

(L3) If asked about any MOD_p-axiom he must say true.

(L4) If asked about any formula from ($\neg$PHP$_n$) he must say true.

The game runs for t rounds of questions and Liar wins if he can always answer while obeying the rules. Otherwise Prover wins.

Lemma 2.1 *For any $d \geq 2$, $n \geq 1$ and $s \geq 1$. If there is a size s $LK_d(MOD_p)$ refutation of ($\neg PHP_n$) then Prover has a winning strategy for game*

$$G(d + O(1), n, O(\log s)) \ .$$

Proof :

It is well-known that LK-proofs (or Frege proofs) can be put into a form of balanced tree with only a polynomial increase in size and a constant increase in the depth (cf. [18, 19]). In particular, the hypothesis of the lemma implies that there is a size $s^{O(1)}$ refutation π of ($\neg$PHP$_n$) in $LK_{d+O(1)}(MOD_p)$ that is in a form of tree whose depth is $O(\log s)$.

The Prover will attempt - by asking Liar suitable questions - to built a path of sequents $Z_1, Z_2, \ldots$ in π such that

- Z_1 is the end-sequent of π, i.e. the empty sequent.
- Z_{i+1} is one of the hypothesis of the inference yielding Z_i.
- If Z_i is $\Gamma \to \Delta$ then Prover asked all formulas in Γ, Δ and Liar asserted that all formulas in Γ are true and all formulas in Δ are false.

Assume $Z_1, \ldots, Z_i$ has been constructed. Next Prover's move depends on the type of inference yielding Z_i:

- Structural rules: Prover asks no questions and just takes for Z_{i+1} the hypothesis of the inference.
- Cut rule: Prover asks about the truth value of the cut formula, say φ, and if Liar asserts it to be true, Prover takes for Z_{i+1} the hypothesis of the inference having φ in the antecedent, otherwise it takes the hypothesis with φ in the succedent.
- A $\neg$ introduction rule: if $\neg\varphi$ was the formula introduced, Prover asks φ and takes for Z_{i+1} the unique hypothesis of the inference.
- The $\bigvee : right$ introduction rule: if the principal formula was $\varphi = \bigvee_{i \leq u} \varphi_i$ and the minor formula φ_j Prover already asked φ in an earlier round and got answer false. He now asks φ_j and takes for Z_{i+1} the unique hypothesis of the inference.
- The $\bigvee : left$ introduction rule: if the principal formula was $\varphi = \bigvee_{i \leq u} \varphi_i$ Prover already asked φ in an earlier round and got answer true. She now asks Liar to witness this answer by some φ_j and then takes for Z_{i+1} the hypothesis with the minor formula φ_j in the antecedent.

This process either causes Liar to lose or otherwise arrives at an initial sequent which Liar's answers claim to be false. But that contradicts rules (L1), (L3) or (L4).

q.e.d.

Shallow tree-like refutations of a set of axioms can be used as search trees finding an axiom false under a given assignment: the Liar answers the truth values determined by the assignment (see e.g. the use of such trees in [18, 19]). It was an important insight of Buss and Pudlák [11] that when Liars are allowed not to follow an assignment but are only required to be logically consistent then the minimal length of Prover's winning strategy characterizes the minimal depth of a tree-like refutation (a form of a statement opposite to the lemma also holds as pointed out in [11] in the context of unrestricted Frege systems).

3 Algebraic formulation of ($\neg$PHP$_n$) and a game with polynomials

Let $\mathbf{F}_p[x_{ij} \mid i \in [n+1] \wedge j \in [n]]$ be the ring of polynomials over the finite field $\mathbf{F}_p$ with p elements with the indicated variables. Denote by S_n the ring factored by the ideal generated by all polynomials $x_{ij}^2 - x_{ij}$. Elements of S_n are multi-linear polynomials. Let $S_{n,e}$ be the $\mathbf{F}_p$-vector space of elements of S_n of degree at most e. We shall denote monomials $x_a, \ldots$ where $a, \ldots$ are unordered tuples of variable indices; the monomial is then the product of the corresponding variables.

Beame et al.[6] formulated (the negation of) PHP as the following **($\neg$PHP$_n$)-system** of polynomial equations in S_n:

- $x_{i_1 j} \cdot x_{i_2 j} = 0$, for each $i_1 \neq i_2 \in [n+1]$ and $j \in [n]$.
- $x_{ij_1} \cdot x_{ij_2} = 0$, for each $i \in [n+1]$ and $j_1 \neq j_2 \in [n]$.
- $1 - \sum_{j \in [n]} x_{ij} = 0$, for each $i \in [n+1]$.

The left-hand sides of these equations will be denoted $Q_{i_1,i_2;j}$, $Q_{i;j_1,j_2}$ and Q_i respectively.

The language of rings is a complete language for propositional logic and it is easy to imagine a modification of the G-game to such a language if the answers of Liar have to respect both the sum and the product. The game we are going to define allows only simple questions and requires that sums of two polynomials and products of two monomials are respected.

We shall define the following game $H(e, n, r)$ played by two players Alice and Bob. Alice's role will be similar to that of Prover in the G-game and Bob's to that of Liar. In every round Alice may put to Bob a question of just one type:

(A) She asks Bob to give to a polynomial f from $S_{n,e}$ a value from $\mathbf{F}_p$.

Bob's answers must obey the following rules:

(B0) If asked about a polynomial whose value he gave in an earlier round Bob must answer identically as before.

(B1) He must give to each element $c \in \mathbf{F}_p$ the value c, and to each variable either 0 or 1.

(B2) If he gave values to f, g and $f+g$, the values given to f and g must sum up to the value he gave to $f+g$.

(B3) If he gave values to monomials x_a, x_b and $x_a \cdot x_b$, the product of the values given to x_a and x_b must equal to the value given to $x_a \cdot x_b$.

(B4) He must give value 0 to all polynomials $Q_{i_1,i_2;j}$, $Q_{i;j_1,j_2}$ and Q_i.

The game runs for r rounds and Bob wins if he can answer all questions while obeying the rules. Otherwise Alice wins.

We consider the multiplicativity condition for monomials rather than for polynomials as that more clearly isolates the role of linearity. As is shown in Section 4 the two versions of the multiplicativity condition are essentially equivalent.

In principle Bob's strategy can be adaptive (i.e. his moves depend on the development of the game) or even may depend on Alice. Call a strategy of

Bob **simple** if it is a function B assigning to elements of $S_{n,e}$ values in $\mathbf{F}_p$ and Bob, when asked to evaluate f, answers $B(f)$. We shall abuse the language occasionally and talk about a **simple Bob** rather than a simple strategy for Bob.

4 Five useful protocols for Alice

In this section we describe five simple protocols in which Alice can force Bob to answer various more complicated questions, similar to that of (P2).

Protocol M_0: Assume that Bob asserted that $\sum_{i \le u} f_i \ne 0$. Alice wants to force Bob to assert that $f_j \ne 0$ for some $j \le u$ (or to lose).

She splits the sum into halves and asks Bob to evaluate $\sum_{i \le u/2} f_i$ and $\sum_{i > u/2} f_i$. As he already gave to $\sum_{i \le u} f_i$ a non-zero value, by (B0) and (B2) - unless he quits - Bob must give to at least one of the half-sums a non-zero value. Continuing in a binary search fashion in $\log u$ rounds she forces Bob to assert that $f_j \ne 0$ for some $j \le u$.

Protocol M_1: Assume that Bob gave to some polynomials f, g and $f \cdot g$ values $B(f)$, $B(g)$ and $B(f \cdot g)$ respectively, and that $B(f) \cdot B(g) \ne B(f \cdot g)$. Alice wants to force Bob into a contradiction with the rules.

Alice writes polynomials f and g as $\mathbf{F}_p$-linear combinations of monomials: $f = \sum_{a \in A} c_a x_a$ and $g = \sum_{b \in B} d_b x_b$ with $c_a, d_b \in \mathbf{F}_p$ and x_a, x_b monomials. She splits A into two halves $A = A_0 \dot\cup A_1$, and asks Bob for the values of

$$(\sum_{a \in A_0} c_a x_a)\ ,\ (\sum_{a \in A_0} c_a x_a) \cdot g\ ,\ (\sum_{a \in A_1} c_a x_a)\ ,\ \text{ and } (\sum_{a \in A_1} c_a x_a) \cdot g\ .$$

Unless Bob violates the linearity rule (B2) his answers must satisfy

$$B(\sum_{a \in A_i} c_a x_a) \cdot B(g) \ \ne\ B((\sum_{a \in A_i} c_a x_a) \cdot g)$$

for either $i = 0$ or $i = 1$. Continuing in the binary search fashion Alice forces Bob to assert

$$B(c_a x_a) \cdot B(g) \ \ne\ B(c_a x_a \cdot g)$$

for some monomial x_a. Using (B1) and (B2) she forces

$$B(c_a x_a) = c_a B(x_a) \ \text{ and } \ B(c_a x_a g) = c_a B(x_a g)$$

and hence

$$B(x_a) \cdot B(g) \ \ne\ B(x_a \cdot g)\ .$$

The number of variables is $n^{O(1)}$ and so the number of monomials of degree at most e is $n^{O(e)}$. Hence all this process requires as most $O(e \log n)$ rounds of Alice's questions.

Now she analogously forces Bob to assert

$$B(x_a) \cdot B(x_b) \neq B(x_a \cdot x_b)$$

for some monomial x_b occurring in g, violating thus (B3).

Protocol M_2: Assume that Bob asserted that $\Pi_{i \leq k} f_i \neq 0$ and let $j \leq k$ be arbitrary. Alice wants to force Bob to assert that $f_j \neq 0$ (or to lose).

She asks Bob to state the value of f_j and if Bob says $f_j \neq 0$ she stops. Otherwise the triple f_j, g and $f_j g$ for $g := \Pi_{i \leq k, i \neq j} f_i$ satisfies the hypothesis of protocol M_1 and Alice can win in $O(e \log n)$ rounds.

Protocol M_3: Assume that Bob asserted that $\Pi_{i \leq k} f_i = 0$. Alice wants to force Bob to assert that $f_j = 0$ for some $j \leq k$ (or to lose).

We shall describe the protocol by induction on k. Alice asks first for the value of f_k. If Bob states that $f_k = 0$ she stops. If he states that $f_k \neq 0$ she asks him for the value of $\Pi_{i<k} f_i$. If Bob says that $\Pi_{i<k} f_i = 0$, Alice has - by the induction hypothesis - a way how to solve the task.

If he says that $\Pi_{i<k} f_i \neq 0$ Alice forces him into contradiction using protocol M_1. We may assume that all polynomials f_i are non-constant and thus the induction process takes at most $k \leq e$ steps.

Note that again Alice needed at most $2e + O(e \log n) = O(et)$ rounds in total.

Protocol M_4: Let $g = f^{p-1}$ and assume that Bob gave to g a value different from both $0, 1$. Alice wants to force Bob into a contradiction.

She asks Bob for the value of f and assume Bob states $f = c \in \mathbf{F}_p$. If $c = 0$ Alice uses protocol M_2 to force a contradiction. If $c \neq 0$ Alice asks Bob for values of $f^2, f^3, \ldots, f^{p-1}$ and unless Bob returns values $c^2, c^3, \ldots, c^{p-1}$ she forces him into a contradiction by protocol M_1. But Bob cannot keep up these answers because if he gave to g now the value $c^{p-1} = 1$ he would violate rule (B0).

5 From Prover to Alice and from Bob to Liar

In this section we employ the Razborov - Smolensky method to show that the existence of many simple winning strategies for Bob yields a winning strategy for Liar[1]. The reason to single out simple strategies is that we shall apply the Razborov - Smolensky approximation method in order to move from a G-game to an H-game, by approximating formulas by low degree polynomials with respect to (a set of) Bob's strategies. The approximation process (and hence a

[1]We could have bypassed the G-game and the explicit use of the Razborov - Smolensky method by employing the characterization of the size of $AC^0[p]$ Frege proofs in terms of degree of proofs in the so called Extended Nullstellensatz of [9]. We prefer here a self-contained presentation.

strategy for Alice to be constructed) depends on the set of Bob's strategies we start with and to avoid circularity we restrict to sets containing only (but not necessarily all) simple strategies.

Lemma 5.1 *Let $d \geq 2$, $n \geq 1$ and $t \geq \log n$ be arbitrary and take parameters e, r*

$$e \ := \ ((t^2 + 2t)p)^d \quad and \quad r \ := \ O(et^4) \ .$$

Let P be any strategy for Prover in game $G(d,n,t)$. Let $\Omega_{e,n,r}$ be a non-empty set of simple strategies for Bob in game $H(e,n,r)$.

Then P can be translated into a strategy A for Alice in $H(e,n,r)$ such that the following holds:

- *If*

$$Prob_{B \in \Omega_{e,n,r}}[B \ \ wins\ over\ A\ in\ H(e,n,r)] \ > \ 1 \ - \ 2^{-(t+1)} \qquad (1)$$

then there exists a Liar's strategy L winning over P in $G(d,n,t)$.

Proof :

Let P and $\Omega_{e,n,r}$ be given. Let F be the smallest set of formulas closed under subformulas and containing all possible P's questions according to rule (P1) in all plays of the game $G(d,n,t)$ against all possible Liars. The number of such (P1) questions is at most 2^{t^2} and each has size at most 2^t and so also at most 2^t subformulas. Thus the depth of all formulas in F is at most d and their total number is bounded by 2^{t^2+t}.

We shall use the Razborov - Smolensky method to assign to all formulas $\varphi \in F$ a polynomial $\hat{\varphi} \in S_{n,e}$. However, we shall approximate with respect to Bob's strategies from $\Omega_{e,n,r}$ rather than with respect to all assignments to variables as it is usual.

Fix parameter $\ell := t^2 + 2t$. Put $\hat{x}_e := x_e$, $\widehat{(\neg\varphi)} := 1 - \hat{\varphi}$ and for $\varphi = MOD_{p,i}(\varphi_1, \ldots, \varphi_k)$ define

$$\hat{\varphi} \ := \ 1 \ - \ ((\sum_{j \leq i} \hat{\varphi}_j) - i)^{p-1} \ .$$

For the remaining case $\varphi = \bigvee_{i \in [u]} \varphi_i$ assume that all polynomials $\hat{\varphi}_i$ were already defined. Pick ℓ subsets $J_1, \ldots, J_\ell \subseteq [u]$, independently and uniformly at random (we shall fix them in a moment), and define polynomial

$$p_\varphi(y_1, \ldots, y_u) \ := \ 1 \ - \ \Pi_{j \leq \ell}(1 \ - \ (\sum_{i \in J_j} y_i)^{p-1})$$

and using p_φ put

$$\hat{\varphi} \ := \ p_\varphi(\hat{\varphi}_1, \ldots, \hat{\varphi}_u) \ . \qquad (2)$$

The following claim is easily verified by induction on the depth of φ, using the protocols from Section 4.

Claim 1: *Let $\varphi \in F$ and assume that Bob asserted that $\hat{\varphi} = c \in \mathbf{F}_p$ for some $c \neq 0, 1$. Then Alice can force Bob into a contradiction in $O(e \log n)$ rounds.*

Let $b_i \in \{0,1\}$ be the truth-value of the statement $B(\hat{\varphi}_i) \neq 0$. For $B \in \Omega_{e,n,r}$ we have that

$$\bigvee_{i \in [u]} b_i \;=\; p_\varphi(b_1, \ldots, b_u) \tag{3}$$

with the probability at least $1 - 2^{-\ell}$ (taken over the choices of sets J). Hence we can select specific sets $J_1, \ldots, J_\ell$ such that (3) holds for all but $2^{-\ell} \cdot |\Omega_{e,n,r}|$ simple Bobs from $\Omega_{e,n,r}$. The polynomial $\hat{\varphi}$ in (2) is assumed to have this property.

Define in this way the polynomial $\hat{\varphi}$ for all (at most 2^{t^2+t}) formulas $\varphi \in F$ by induction on the depth $1, 2, \ldots, d$. Each is of degree at most $(\ell(p-1))^d \leq ((t^2+2t)p)^d = e$ and it holds that:

Claim 2: *There is a subset $Err \subseteq \Omega_{e,n,r}$ such that $|Err| \leq 2^{-t}|\Omega_{e,n,r}|$ and such that (3) holds for all disjunctions $\varphi \in F$ and all $B \in \Omega_{e,n,r} \setminus Err$.*

Now we define, using the given strategy P for Prover, a specific strategy A for Alice in $H(e,n,r)$. We transcript P into A a question by question; each question of P may be replaced by a series of questions of Alice.

If P asks according to (P1) what is the value of φ, Alice simply asks for the value of $\hat{\varphi}$. Let $\varphi = \bigvee_{i \in [u]} \varphi_i$ and assume that P asks according to (P2); there are two cases to consider:

(a) φ got value false and P asks for the value of one disjunct φ_j,

(b) φ got value true and P asks for a witness φ_j.

Assume for the case (a) that Bob asserted in an earlier round that $\hat{\varphi} = 0$. Alice asks Bob for the value of $\hat{\varphi}_j$. If he gives $B(\hat{\varphi}_j) = 0$ the simulation of P moves to the next round. If he replies that $B(\hat{\varphi}_j) = 1$, Alice uses first protocol M_2 repeatedly to force Bob to assert

$$1 \;-\; (\sum_{i \in J_v} \hat{\varphi}_i)^{p-1} \;\neq\; 0$$

for all $v \leq \ell$. Then for each v she uses protocol M_4 to force Bob to say that

$$(\sum_{i \in J_v} \hat{\varphi}_i)^{p-1} \;=\; 0$$

and further protocol M_3 to assert that

$$\sum_{i \in J_v} \hat{\varphi}_i \;=\; 0\ . \tag{4}$$

This needs $O(\ell e \log n) = O(t^2 e \log n) = O(et^3)$ rounds.

As $B(\hat{\varphi}_j) = 1$, if Bob uses a strategy $B \in \Omega_{e,n,r} \setminus Err$, the definition of Err guarantees that one of the equations in (4) is false when $\hat{\varphi}_i$'s are evaluated by B:

$$B(\sum_{i \in J_v} \hat{\varphi}_i) \;\neq\; \sum_{i \in J_v} B(\hat{\varphi}_i)\ .$$

This itself is not a violation of rule (B2) but Alice can use this situation and to force Bob to lose. We shall describe her strategy as probabilistic; a deterministic one is obtained by an averaging argument.

Alice splits $J_v = K_0 \dot\cup K_1$ into halves and asks Bob for the values of $\sum_{i\in K_0} \hat\varphi_i$ and $\sum_{i\in K_1} \hat\varphi_i$. Unless he violates (B2) his answers must sum up to $B(\sum_{i\in J_v} \hat\varphi_i)$. Hence for $k = 0$ or $k = 1$

$$B(\sum_{i\in K_k} \hat\varphi_i) \;\neq\; \sum_{i\in K_k} B(\hat\varphi_i) \;.$$

Alice guesses for which k this happens and then proceeds analogously with $\sum_{i\in K_k} \hat\varphi_i$, splitting it into halves, asking Bob for the values, etc. If she always guesses right then in t steps (as the size of the sums is bounded by 2^t) she will reduce the sums to one term and will win. Alice has the probability at least 2^{-t} to make the right choices. She does not know a priori which of the ℓ sums $\sum_{i\in J_v} \hat\varphi_i$ to use so she must try all. This takes $O(\ell t) = O(t^3)$ rounds.

There are at most t simulations of a (P2a) question in the G-game but Alice needs to employ the random strategy above only once when the case $B(\hat\varphi_j) = 1$ occurs, and then her probability of success is at least 2^{-t}. By averaging there are fixed choices that Alice can make, yielding this success probability outside of Err. In particular, for a random $B \in \Omega_{e,n,r} \setminus Err$, if Alice uses these choices then either B must give to $\hat\varphi_j$ value 0 or Alice wins with the probability at least 2^{-t}. We shall describe this situation below by the phrase that the *(P2a) simulation succeeded*.

Assume for the case (b) that Bob answered earlier that $\hat\varphi = 1$ and hence also that

$$\Pi_{j\leq\ell}(1 \;-\; (\sum_{i\in J_j} \hat\varphi_i)^{p-1}) \;=\; 0.$$

Alice uses protocols M_3 and M_4 to force Bob to state that $\sum_{i\in J_v} \hat\varphi_i = 1$ for some $v \leq \ell$. This uses $O(e \log n) = O(et)$ rounds. Then she uses protocol M_0 to force Bob to say that $\hat\varphi_j \neq 0$ for some $j \in J_v$ and by Claim 1 the value has to be 1 ($O(e \log n) = O(et)$ rounds are used in Claim 1). The number of formulas φ_i is bounded by the size of φ, i.e. by 2^t, and so this uses at most t rounds in protocol M_0, i.e. still $O(et)$ in total.

This describes the strategy A.

By Claim 2 with the probability at least $1 - 2^{-t}$ a random $B \in \Omega_{e,n,r}$ is outside Err, and for these Alice's simulations of (P2a) questions succeed with the probability at least 2^{-t}. Thus the inequality (1) from the hypothesis of the lemma implies that there is at least one $B \in \Omega_{e,n,r} \setminus Err$ winning over the particular Alice's strategy A and for which A's simulations of (P2a) questions succeed.

Use B to define a strategy L for Liar in the original game $G(d, n, t)$ simply by giving to φ the truth value $B(\hat\varphi)$ when asked a (P1) type question, and giving a witness φ_j constructed in the case (b) above when asked a (P2b) type question.

From the construction of A (and rules for Bob) it follows that L satisfies the rules for Liar. In particular, by (B4) all polynomials from the ($\neg$PHP$_n$)-system get 0 by B and so all axioms of ($\neg$PHP$_n$) get by L value true.

Note that one question of P is transcribed into at most $O(et^3)$ Alice's questions. Hence in every play of the H-game transcribing a play of the G-game there are in total at most $r = O(et^4)$ rounds.

q.e.d.

6 A general reduction to a search problem

The reduction of the lengths-of-proofs problem to a question about the H-games in Sections 2 - 5 is not specific to ($\neg$PHP$_n$) and works in a fairly general situation that we shall describe now. Then we reduce the proof complexity problem further to a question about the computational complexity of a certain task involving computations with search trees.

The only specific thing in the ($\neg$PHP$_n$) case is the transcription of the axioms of ($\neg$PHP$_n$) into the ($\neg$PHP$_n$) polynomial system in Section 3. This is not a mere mechanical translation from DeMorgan language into the language of rings (in that the axioms $\bigvee_{j\in[n]} p_{ij}$ would translate into polynomials of degree about n and not into degree 1 polynomials Q_i). In order to avoid inevitable technicalities when trying to define suitable translations from a general set of axioms to a polynomial system we simply take as our starting point an unsolvable system of polynomial equations of a constant degree. The truth value of an equation $f(x_1,\ldots,x_m) = 0$ for Boolean variables x_i, f a degree $O(1)$ polynomial over $\mathbf{F}_p$, can be defined by a depth 2, size $m^{O(1)}$ $AC^0[p]$ formula. Namely, writing f as an $\mathbf{F}_p$-linear combination $\sum_{a\in A} c_a x_a$ of monomials x_a, with $c_a \in \{1,\ldots,p-1\}$, consider the formula

$$\varphi \ := \ MOD_{p,0}(\psi_1,\ldots,\psi_k) \tag{5}$$

where $k = \sum_{a\in A} c_a$ and ψ_i's are conjunctions of variables corresponding to monomials from f, each monomial x_a being represented c_a-times. Clearly[2] φ represents the truth value of $f = 0$ on Boolean variables. The polynomial system can thus be also thought of as an unsatisfiable set of $AC^0[p]$ formulas and we can speak about its $LK_d(MOD_p)$ refutations.

We shall now consider the following general set-up. For $n = 1, 2, \ldots$ let $\mathcal{F}_n$ be a sequence of sets of polynomials over $\mathbf{F}_p$ in variables $Var(\mathcal{F}_n)$. We shall assume that:

1. polynomials in sets $\mathcal{F}_n$ have $O(1)$ degree,
2. the size of both $\mathcal{F}_n$ and $Var(\mathcal{F}_n)$ is $n^{O(1)}$,

[2]Instead of assuming degree $O(1)$ it would suffice to assume that f is an $\mathbf{F}_p$-linear combination of polynomially many monomials.

3. the polynomial system

$$f = 0 , \quad \text{for} \quad f \in \mathcal{F}_n$$

contains equations $x^2 - x = 0$ for all $x \in Var(\mathcal{F}_n)$ and is unsolvable in $\mathbf{F}_p$.

Let $S^{\mathcal{F}}_{n,e}$ be the $\mathbf{F}_p$ - vector space of multi-linear polynomials in variables of $\mathcal{F}_n$ and of degree at most e.

We want to replace games and strategies considered in previous sections by a more direct computational model, namely that of search trees. Define an $S^{\mathcal{F}}_{n,e}$ **- search tree** T to be a p-ary tree whose inner nodes (non-leaves) are labelled by polynomials from $S^{\mathcal{F}}_{n,e}$, the p edges leaving a node labelled by g are labelled by $g = 0, g = 1, \ldots, g = p-1$, and leaves are labelled by elements of a set X.

Any function $B : S^{\mathcal{F}}_{n,e} \to \mathbf{F}_p$ determines a path $P_T(B)$ in T consisting of edges labelled by $g = B(g)$ and thus it also determines an element of X: the label of the unique leaf on $P_T(B)$. Hence T defines a function assigning to any map $B : S^{\mathcal{F}}_{n,e} \to \mathbf{F}_p$ an element of X to be denoted $T(B)$.

Let $Error^{\mathcal{F}}_{n,e}$ be the set of pairs and triples of the form $(B1, c)$ for $c \in \mathbf{F}_p$ or $(B1, x)$ for $x \in Var(\mathcal{F}_n)$, $(B2, f, g)$, $(B3, x_a, x_b)$ or $(B4, f)$ for $f \in \mathcal{F}_n$, with f, g, x_a, x_b of degree at most e. These are intended to indicate what instance of which rule did Bob violate. We say that $(B1, c)$ is an **error for** B iff $B(c) \neq c$, $(B1, x)$ is an error for B iff $B(x) \neq 0, 1$, and similarly for the other pairs and triples[3].

In the following statement we talk about refutations of equations $f = 0$, $f \in \mathcal{F}_n$. As pointed out earlier, we can view them also as depth 2, polynomial size formulas with $MOD_{p,0}$ connectives and hence it makes a prefect sense to talk about their $LK_d(MOD_p)$-refutations.

The reductions of Sections 2 - 5 used the example of ($\neg$PHP$_n$) (see the beginning of Section 2) but nothing specific to it was used. Hence we can employ the reductions to derive the following general statement. In it we replace degree e by (bigger) r in order to avoid the need to define here the relation between them implicit in Lemma 5.1.

Theorem 6.1 *Let $r = r(n) \geq (\log n)^{\omega(1)}$ be a function and let $\mathcal{F}_n$ be sets of polynomials obeying the restrictions 1., 2. and 3. listed above.*

Then for every $d \geq 2$ there are $\epsilon_d > 0$ and $n_d \geq 1$ such that for an arbitrary non-empty set $\Omega_{\mathcal{F}_n,r}$ of maps from $S^{\mathcal{F}}_{n,r}$ to $\mathbf{F}_p$ the following implication (I) holds for all $n \geq n_d$ and all $0 < \epsilon \leq \epsilon_d$:

(I) If for every $S^{\mathcal{F}}_{n,r}$ - search tree T of depth r and with leaves labelled by elements of $Error^{\mathcal{F}}_{n,r}$ it holds that

$$Prob_{B \in \Omega_{\mathcal{F}_n,r}}[\, T(B) \text{ is not an error for } B \,] \; > \; 1 - 2^{-r^{\epsilon}} \tag{6}$$

then $LK_d(MOD_p)$ does not refute the set of formulas $f = 0$, $f \in \mathcal{F}_n$, by a proof of size less than $2^{\Omega(r^{\epsilon})}$.

[3] We ignore errors for (B0) as that rule cannot be violated by a simple Bob and hence search trees do not need to ask anything twice on any path.

Proof :

Assume that $LK_d(MOD_p)$ does refute the set of formulas $f = 0$, $f \in \mathcal{F}_n$, by a proof of size $s = s(n)$. By Lemma 2.1 Prover has a winning strategy P for game $G(d+c, n, t)$, where $t = t(n) = O(\log s)$ and c is an absolute constant.

Put $\epsilon_d := \frac{1}{2(d+c)+5}$ and let $0 < \epsilon \leq \epsilon_d$. If it were that $t + 1 \leq r^\epsilon$ then the parameters e', r' of the game $H(e', n, r')$ constructed in Lemma 5.1 satisfy $e' \leq r' < r$ and, in particular, the game is an $H(r, n, r)$ game.

The strategy A defined in Lemma 5.1 for the game defines an $S^{\mathcal{F}}_{n,r}$ - search tree T of depth r and with leaves labelled by elements of $Error^{\mathcal{F}}_{n,r}$ in a natural way: a path in T corresponds to possible answers of a simple Bob and the path stops as soon as a violation of one of the rules (B1)-(B4) occurs (rule (B0) cannot be broken by a simple Bob). The label of the resulting leaf is the instance of the rule that was broken (if a violation did not occur we use any element of $Error^{\mathcal{F}}_{n,r}$).

Assume that $\Omega_{\mathcal{F}_n,r}$ is a set of simple Bobs for which the inequality (6) holds. Then also the inequality (1) from Lemma 5.1 holds and thus by that lemma there is a strategy L for Liar that wins over P in the original G-game. That is a contradiction and thus $s \geq 2^{\Omega(r^\epsilon)}$.

q.e.d.

To conclude the paper let us discuss informally the construction underlying Lemma 5.1 and Theorem 6.1. In particular, we see these formal statements as templates for a possible variety of analogous reductions, and it is not clear which one - if any - will be eventually useful.

The strategy A is constructed in Lemma 5.1 by a randomized process from strategy P and from set $\Omega_{e,n,r}$. Let us call the class of all strategies A that can occur in this way the class of $(P, \Omega_{e,n,r})$-*generated strategies*. One such class contains only a few of all possible Alice's strategies. Moreover, we can pick $\Omega_{e,n,r}$ depending on P. Hence one can weaken the hypothesis in these statements and, for example, Theorem 6.1 could be reformulated as follows:

- Let $r = r(n) \geq (\log n)^{\omega(1)}$ be a function and let $\mathcal{F}_n$ be sets of polynomials obeying the restrictions 1., 2. and 3. listed above.

  Then for every $d \geq 2$ there are $\epsilon_d > 0$ and $n_d \geq 1$ such that the following holds:

  If for every Prover's strategy P for game $G(d, n, r^{\Omega(1)})$ there exists a non-empty set $\Omega_{\mathcal{F}_n,r}(P)$ of maps from $S^{\mathcal{F}}_{n,r}$ to $\mathbf{F}_p$ then the following implication (I') holds for all $n \geq n_d$ and all $0 < \epsilon \leq \epsilon_d$:

  (I') If for every $S^{\mathcal{F}}_{n,r}$ - search tree T of depth r and with leaves labelled by elements of $Error^{\mathcal{F}}_{n,r}$ originating from a $(P, \Omega_{\mathcal{F}_n,r}(P))$-generated A it holds that

$$\mathrm{P}rob_{B \in \Omega_{\mathcal{F}_n,r}(P)}[\ T(B) \text{ is not an error for } B\ ] \ > \ 1 - 2^{-r^\epsilon} \qquad (7)$$

then $LK_d(MOD_p)$ does not refute the set of formulas $f = 0$, $f \in \mathcal{F}_n$, by a proof of size less than $2^{\Omega(r^\epsilon)}$.

This formulation stains the combinatorially clean original formulation by a reference to P but (I') may be a weaker hypothesis to arrange.

Another issue is the discouragingly high probability required in (1) and (6). This is due solely by Alice's simulation of the (P2a) move of P. At that point she found $\ell \leq O(t^2) \leq e \leq r$ sets K, $|K| \leq 2^t$, of degree e polynomials such that for one of them B fails linearity:

$$B(\sum_{i \in K} g_i) \neq \sum_{i \in K} B(g_i) \tag{8}$$

and her strategy worked up to this point for all $B \notin Err$ (as long as P was a winning strategy for the Prover). Getting from this situation to a violation of rule (B2) costs her the drop of the success probability by the multiplicative factor 2^{-t}. Hence we could redefine the rules for the H-game and, in particular, the error sets $Error^{\mathcal{F}}_{n,e}$ for the search problems to be solved by the trees, and include that situation (i.e. A producing ℓ sets K such that one of them satisfies (8)) among the stopping Bob's errors. Let us call $*Error^{\mathcal{F}}_{n,e}$ the set of errors with this new type of an error added. Then we could reformulate Theorem 6.1 differently as follows:

- Let $r = r(n) \geq (\log n)^{\omega(1)}$ be a function and let $\mathcal{F}_n$ be sets of polynomials obeying the restrictions 1., 2. and 3. listed above.

  Then for every $d \geq 2$ there are $\epsilon_d > 0$ and $n_d \geq 1$ such that for an arbitrary non-empty set $\Omega_{\mathcal{F}_n,r}$ of maps from $S^{\mathcal{F}}_{n,r}$ to $\mathbf{F}_p$ the following implication (I") holds for all $n \geq n_d$ and all $0 < \epsilon \leq \epsilon_d$:

  (I") If for every $S^{\mathcal{F}}_{n,r}$ - search tree T of depth r and with leaves labelled by elements of $*Error^{\mathcal{F}}_{n,r}$ it holds that

  $$\mathrm{P}rob_{B \in \Omega_{\mathcal{F}_n,r}}[\ T(B) \text{ is not an error for } B\ ]\ >\ 2^{-r^\epsilon} \tag{9}$$

  then $LK_d(MOD_p)$ does not refute the set of formulas $f = 0$, $f \in \mathcal{F}_n$, by a proof of size less than $2^{\Omega(r^\epsilon)}$.

Let us stress that the culprit property is the linearity by observing that simple Bobs can be without a loss of generality assumed to satisfy all rules except possibly (B2). First, having B we can define B' by correcting all values of B that violate rules (B1) or (B4). If B' is asked by Alice for one of these new values, the original B would lose. Hence B' is as good as B against any A.

Then define B'' by giving to every monomial $x_a = \Pi_i x_i$ the value $\Pi_i B'(x_i)$. Enhance any A to vigilant A^* that whenever she asks for the value of a monomial, she asks also for the values of all its variables (this enlarges the number of round e-times at most). Clearly, B'' fares as well as B' against a vigilant A^*.

Finally, let us remark that it would be interesting and possibly quite useful to modify the construction so that adaptive Bobs are allowed.

Acknowledgements.

I thank L. Kolodziejczyk (Warsaw), S. Müller (Tokyo), P. Pudlák (Prague) and N. Thapen (Prague) for critical comments and discussions. I am also indebted to the anonymous referee for helpful suggestions.

References

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

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

[3] M. Ajtai, Parity and the pigeonhole principle, in: Feasible Mathematics, eds. S. R. Buss and P. J. Scott, Birkhauser, (1990), pp.1-24.

[4] M. Ajtai, The independence of the modulo p counting principles, in: *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, (1994), pp.402-411. ACM Press.

[5] M. Ajtai, Symmetric Systems of Linear Equations modulo p, in: Electronic Colloquium on Computational Complexity (ECCC), TR94-015, (1994).

[6] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák: Lower bounds on Hilbert's Nullstellensatz and propositional proofs, *Proceedings of the London Mathematical Society*, **(3) 73**, (1996), pp.1-26.

[7] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, P. Pudlák, and A. Woods: Exponential lower bounds for the pigeonhole principle, in: *Annual ACM Symp. on Theory of Computing*, (1992), pp.200-220.

[8] S. R. Buss, Lower Bounds on Nullstellensatz Proofs via Designs, in: *Proof Complexity and Feasible Arithmetics*, eds. S. R. Buss and P. Beame, American Mathematical Society, Providence, RI, (1998), pp. 59-71.

[9] S. R. Buss, R. Impagliazzo, J. Krajíček, P. Pudlák, A. A. Razborov, and J. Sgall: Proof complexity in algebraic systems and bounded depth Frege systems with modular counting, *Computational Complexity*, **6(3)**, (1996/1997), pp.256-298.

[10] S. R. Buss, L. A. Kolodziejczyk, and K. Zdanowski: Collapsing modular counting in bounded arithmetic and constant depth propositional proofs, preprint, (2013).

[11] S. R. Buss and P. Pudlák, How to lie without being (easily) convicted and the lengths of proofs in propositional calculus, in Computer Science Logic'94, Pacholski and Tiuryn eds., Springer-Verlag, LNCS **933**, (1995), pp.151-162.

[12] M. Clegg, J. Edmonds, and R. Impagliazzo, Using the Groebner basis algorithm to find proofs of unsatisfiability, in: *Proc. 28th Annual ACM Symp. on Theory of Computing*, (1996), pp. 174-183. ACM Press.

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

[14] M. Furst, J. B. Saxe, and M. Sipser, M. Parity, circuits and the polynomial-time hierarchy, *Math. Systems Theory*, (1984), **17**: 13–27.

[15] J. Hastad, Almost optimal lower bounds for small depth circuits. in: *Randomness and Computation*, ed. S.Micali, Ser.Adv.Comp.Res., **5**, (1989), pp.143-170. JAI Pres.

[16] R. Impagliazzo, P. Pudlak, and J. Sgall, Lower bounds for the polynomial calculus and the Groebner basis algorithm, *Comput. Complexity*, **8(2)**, (1999), pp.127-144.

[17] R. Impagliazzo and N. Segerlind, Counting axioms do not polynomially simulate counting gates, in: *Proc. IEEE 42nd Annual Symp. on Foundation of Computer Science*, (2001), pp. 200-209.

[18] J. Krajíček, Lower bounds to the size of constant-depth propositional proofs, *Journal of Symbolic Logic*, **59(1)**, (1994), pp.73-86.

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

[20] 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.

[21] J. Krajíček, On the degree of ideal membership proofs from uniform families of polynomials over a finite field, *Illinois J. of Mathematics*, **45(1)**, (2001), pp.41-73.

[22] J. Krajíček, *Forcing with random variables and proof complexity*, London Mathematical Society Lecture Note Series, No. **382**, Cambridge University Press, (2011).

[23] 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.

[24] A. Maciel and T. Pitassi, Towards lower bounds for bounded-depth Frege proofs with modular connectives, in: *Proof Complexity and Feasible Arithmetics*, P. Beame and S. Buss, eds., DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. **39**, pp. 195-227, American Mathematical Society, (1998).

[25] A. Maciel and T. Pitassi, A Conditional Lower Bound for a System of Constant-Depth Proofs with Modular Connectives, in: Proc. of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS 06), IEEE Computer Society Press, (August 2006).

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

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

[28] A. A. Razborov, Lower bounds on the size of bounded depth networks over a complete basis with logical addition, *Matem. Zametki*, **41(4)**, (1987), 598-607.

[29] A. A. Razborov, Lower Bounds for the Polynomial Calculus, *Computational Complexity*, **7(4)**, (1998), pp.291-324.

[30] 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.

[31] A. Yao, Separating the polynomial-time hierarchy by oracles, in: *Proc. 26th Ann. IEEE Symp. on Found. of Comp. Sci.*, (1985), pp. 1-10.

Mailing address:

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`