Electronic Colloquium on Computational Complexity, Report No. 11 (2001)

Algebraic proof systems over formulas

Dima Grigoriev* Edward A. Hirsch†

December 2000

Abstract

We introduce two algebraic propositional proof systems $\mathcal{F}$-$\mathcal{NS}$ and $\mathcal{F}$-$\mathcal{PC}$. The main difference of our systems from (customary) Nullstellensatz and Polynomial Calculus is that the polynomials are represented as arbitrary formulas (rather than sums of monomials). Short proofs of Tseitin's tautologies in the constant-depth version of $\mathcal{F}$-$\mathcal{NS}$ provide an exponential separation between this system and Polynomial Calculus.

We prove that $\mathcal{F}$-$\mathcal{NS}$ (and hence $\mathcal{F}$-$\mathcal{PC}$) polynomially simulates Frege systems, and that the constant-depth version of $\mathcal{F}$-$\mathcal{PC}$ over finite field polynomially simulates constant-depth Frege systems with modular counting. We also present a short constant-depth $\mathcal{F}$-$\mathcal{PC}$ (in fact, $\mathcal{F}$-$\mathcal{NS}$) proof of the propositional pigeon-hole principle. Finally, we introduce several extensions of our systems and pose numerous open questions.

1 Introduction

A (Cook–Reckhow) *proof system* [CR79] for a language L is a polynomial-time computable function mapping strings in some alphabet onto L. If there would be a proof system Π for a co-**NP**-hard language such that for every $x \in L$, the shortest proof π of x (i.e., the shortest string π such that $\Pi(\pi) = x$) had size polynomial in the size of x, then we would have **NP** = co-**NP**.

A proof system Π_1 *polynomially simulates* a proof system Π_2 iff there is a polynomial-time computable function f mapping every Π_2 proof π to a Π_1 proof of the same element of L, i.e., $\Pi_1(f(\pi)) = \Pi_2(\pi)$.

A *propositional proof system* is a proof system for the co-**NP**-complete language **TAUT** of all Boolean tautologies. Since this language is in co-**NP**, any proof system for a co-**NP**-hard language L can be considered as a propositional proof system. However, note that one needs to fix a concrete reduction of **TAUT** to L before proving, e.g., that such a system does (not) polynomially simulate another propositional proof system.

Algebraic proof systems played a significant role in propositional proof complexity during the past decade. The two most popular systems are Nullstellensatz ($\mathcal{NS}$) [BIK+96] and Polynomial Calculus ($\mathcal{PC}$) [CEI96]. These are proof systems for the co-**NP**-hard problem of unsolvability of a system of polynomial equations: We are given several polynomials over a field $\mathbb{F}$ and asked whether these polynomials have no common roots in the algebraic closure of $\mathbb{F}$. The polynomials

*IRMAR, Université de Rennes, Campus de Beaulieu, 35042 Rennes, cedex France. Email: dima@maths.univ-rennes1.fr. Web: http://www.maths.univ-rennes1.fr/~dima.

†Steklov Institute of Mathematics at St.Petersburg, 27 Fontanka, 191011 St.Petersburg, Russia. Email: hirsch@pdmi.ras.ru. Web: http://logic.pdmi.ras.ru/~hirsch. Supported by INTAS Fellowship YSF 99-4044 and grants from NATO and RFBR.

ISSN 1433-8092

are represented as sums of monomials $cxy\ldots z$, where $x, y, \ldots, z$ are variables, and c is a constant (represented in binary).

To see that this problem is co-**NP**-hard, note that a Boolean formula Φ in CNF in n variables $x_1, \ldots, x_n$ can be easily translated into polynomials $F_1, \ldots, F_k$ such that the polynomials

$$F_1, \quad \ldots, \quad F_k, \quad x_1^2 - x_1, \quad \ldots, \quad x_n^2 - x_n$$

have no common roots iff Φ is unsatisfiable. Namely, let $x_{j_1}, \ldots, x_{j_m}$ be the variables occuring in the i-th clause C_i of Φ. Then $F_i = l_1 \ldots l_m$ where $l_t = x_{j_t}$ if x_{j_t} occurs positively in C_i and $l_t = (1 - x_{j_t})$ otherwise.

In $\mathcal{PC}$, one starts with a system of polynomial equations (i.e., with a set of polynomials treated as axioms) and derives new polynomials using two rules

$$\frac{P_1;\ P_2}{P_1 + P_2} \qquad \text{and} \qquad \frac{P}{P \cdot Q}.$$

I.e., one can take a sum[1] of two already derived polynomials P_1 and P_2, or multiply an already derived polynomial P by an arbitrary polynomial Q. The goal is to derive the polynomial 1, because this would mean that Φ is unsatisfiable.

In $\mathcal{NS}$, a proof of $\{F_1, \ldots, F_m\}$ is a set of polynomials $\{G_1, \ldots, G_m\}$ such that $\sum_i F_i G_i = 1$. Any such proof can be translated into $\mathcal{PC}$ in a natural way. However, the translation in the opposite direction is not possible: there is a sequence of tautologies having polynomial-size $\mathcal{PC}$ proofs but no polynomial-size $\mathcal{NS}$ proofs [CEI96]. In fact, $\mathcal{NS}$ is equivalent to the tree-like version of $\mathcal{PC}$ [BIK$^+$97].

It is known that both $\mathcal{PC}$ and $\mathcal{NS}$ are sound and complete, but they are not polynomially bounded. Namely, one can prove a linear lower bound on the maximum degree of the intermediate polynomials (this is done in [Raz98] for axiom polynomials of logarithmic degree, and in [BGIP99] for axiom polynomials of constant degree; see also [AR00]). Then by [IPS99, Theorem 6.2] one obtains an exponential lower bound on the total number of monomials in the proof.

Pitassi [Pit97] considered a variant of $\mathcal{PC}$ where polynomials are represented as formulas (i.e., terms) and not as sums of monomials. The resulting system is still sound and complete, however, it is not a Cook–Reckhow proof system because no polynomial-time deterministic procedure is known that could decide whether an inference rule is applied in a right way. (Note: if we take a sum of two polynomials P and Q in this system, we get not just the term $(P + Q)$—the system would be incomplete in this way—but we can get an arbitrary term representing the same polynomial).

In this paper, we augment this system by primitive rules that help to demonstrate that two terms represent the same polynomial: associativity, commutativity, distributivity, etc. (See **Subsection 2.1**). These rules basically mean that we work in a commutative ring; throughout this paper we call them *the primitive rules*. We also require the next formula to be derived using either the primitive rules or as a formal combination $(P_1 + P_2)$ (resp., $(P \cdot Q)$) of two already derived polynomials P_1 and P_2 (resp., of an already derived polynomial P and an arbitrary polynomial Q). Therefore, we replace every "hard" step like

$$\frac{(x + y) \cdot (x - y)}{x^2 - y^2}$$

[1] Usually, an arbitrary linear combination is allowed, but clearly it can be replaced by two multiplications and one addition.

by a sequence of "primitive" (polynomial-time verifiable) steps like

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(x+y)\cdot(x-y)}{x\cdot(x-y)+y\cdot(x-y)}}{(x-y)\cdot x+(x-y)\cdot y}}{x^2-yx+xy-y^2}}{x^2-xy+xy-y^2}}{x^2+(-1+1)xy-y^2}}{x^2+0-y^2}}{x^2-y^2}.$$

Although it may be still hard to derive a different term representing an already derived polynomial, we can show that the power of the new calculus is sufficient to polynomially simulate Frege systems. Namely, every proof in a Frege system can be translated into a proof in our new system (which we call $\mathcal{F}$-$\mathcal{PC}$) with at most polynomial increase in size (see **Section 3**).

In **Section 4** we show that the tree-like version of $\mathcal{F}$-$\mathcal{PC}$ is polynomially equivalent to $\mathcal{F}$-$\mathcal{NS}$, an analog of $\mathcal{NS}$ in which G_i's may be represented as arbitrary formulas and the equality $\sum_i F_i G_i = 1$ must be proved using the primitive rules only (this system is defined in **Subsection 2.2**). It is known that Frege systems do not lose their power even if restricted to tree-like proofs (see, e.g., [Kra95]). Since our polynomial simulation of Frege systems by $\mathcal{F}$-$\mathcal{PC}$ converts tree-like proofs into tree-like ones, it follows that $\mathcal{F}$-$\mathcal{NS}$ polynomially simulates Frege systems.

We then consider the constant-depth version of $\mathcal{F}$-$\mathcal{PC}$ (i.e., the depth of every formula in the proof is bounded by a constant, see **Subsection 2.3** for definitions, cf. [GR00] where lower bounds for depth-3 arithmetic formulas were established) and restrict our attention to finite fields. It turns out (see **Section 5**) that this system polynomially simulates constant-depth Frege systems with modular gates. We follow [Kra95] in the definition of the latter system. This system has already been considered in [BIK$^+$97] in connection to $\mathcal{NS}$ with a constant number of levels of extension axioms.

We introduce also two extensions of $\mathcal{F}$-$\mathcal{PC}$: one extension involving polynomial inequalities (see **Subsection 2.4**) and another one allowing to deduce a polynomial when some its power is deduced (see **Subsection 2.5**). We illustrate a possible application of this "radical rule" by employing the trick of Rabinowitsch [vdW31, Part 2, Chapter 11] to transform a derivation of 1 from a set $\{1 - zF, F_1, \ldots, F_m\}$ into a derivation of F from the set $\{F_1, \ldots, F_m\}$.

In **Section 6** we present a short $\mathcal{F}$-$\mathcal{PC}$ proof (over $\mathbb{Q}$) of the propositional pigeon-hole principle (PHP) as well as for the subset sum problem. There is nothing surprising in the existence of polynomial-size $\mathcal{F}$-$\mathcal{PC}$ proofs of PHP, because PHP has polynomial-size Frege proofs [Bus87] and $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates Frege systems. However, Buss' proof [Bus87] involves a complicated construction of addition circuits while our proof is very simple and intuitive (actually, we employ the ability of $\mathcal{F}$-$\mathcal{PC}$ to count). Our proof also has depth bounded by a constant, while constant-depth Frege systems do not have polynomial-size proofs of PHP [KPW95, PBI93]. In fact, our proof can be conducted in constant-depth $\mathcal{F}$-$\mathcal{NS}$.

The results of Section 6, however, do not suffice to prove an exponential gap between the lengths of proofs in $\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{PC}$ as *propositional* proof systems (see the discussion in the end of Section 6). In **Section 7** we demonstrate this gap for Tseitin's tautologies. In fact, we demonstrate the gap between $\mathcal{PC}$ and constant-depth $\mathcal{F}$-$\mathcal{NS}$, the weakest system among the ones introduced in this paper.

2 The systems $\mathcal{F}$-$\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{NS}$ and their extensions

In this section we introduce the two systems we study in this paper, and discuss several their extensions.

2.1 $\mathcal{F}$-$\mathcal{PC}$

The objects of our system $\mathcal{F}$-$\mathcal{PC}$ are algebraic formulas. Formally, algebraic formulas are the members of the smallest set satisfying the following conditions:

1. Constants (denoted by 1, -1, 0, etc.) are formulas.
2. Variables are formulas.
3. If P and Q are formulas, then the terms $(P + Q)$ and $(P \cdot Q)$ are formulas.

Constants and variables range over $\mathbb{Q}$ or over any finite field $\mathbb{Z}_p$. Sometimes when speaking about algebraic formulas we will refer to them as polynomials.

Similarly to $\mathcal{PC}$, the two basic rules are

$$\frac{P_1;\ P_2}{(P_1 + P_2)} \qquad \text{and} \qquad \frac{P}{(P \cdot Q)}. \tag{2.1}$$

Note that $(P_1 + P_2)$ and $(P \cdot Q)$ here are terms (i.e., formal combinations): no actual addition or multiplication is done. Since these rules are able to produce only larger formulas, to derive the formula 1 (which is our goal) we need also some (invertible) simplification rules (called *the primitive rules* throughout this paper) yielding associativity, commutativity, distributivity, etc. We allow to apply the primitive rules (but not (2.1)!) to any subterms of the derived formulas, for example, the first rule below can be applied as $\dfrac{(A + ((P + Q) + R))}{(A + (P + (Q + R)))}$. Here is the list of these rules:

$$\frac{((P + Q) + R)}{(P + (Q + R))}, \qquad \frac{(P + (Q + R))}{((P + Q) + R)}, \tag{2.2}$$

$$\frac{((P \cdot Q) \cdot R)}{(P \cdot (Q \cdot R))}, \qquad \frac{(P \cdot (Q \cdot R))}{((P \cdot Q) \cdot R)}, \tag{2.3}$$

$$\frac{((P + Q) \cdot R)}{((P \cdot R) + (Q \cdot R))}, \qquad \frac{((P \cdot R) + (Q \cdot R))}{((P + Q) \cdot R)}, \tag{2.4}$$

$$\frac{(P \cdot 1)}{P}, \qquad \frac{P}{(P \cdot 1)}, \tag{2.5}$$

$$\frac{(P \cdot 0)}{0}, \qquad \frac{0}{(P \cdot 0)}. \tag{2.6}$$

We also allow to replace a subterm P containing only constants by its value c_P (for example, $(-1 + 1)$ simplifies to 0) and vice versa:

$$\frac{P}{c_P}, \qquad \frac{c_P}{P}. \tag{2.7}$$

An $\mathcal{F}$-$\mathcal{PC}$ *proof*[2] of a set $\{F_1, \ldots, F_m\}$ of algebraic formulas is the derivation of the formula 1 from the axioms $F_1, \ldots, F_m$ using the rules (2.1)–(2.7).

As we already mentioned, to consider $\mathcal{F}$-$\mathcal{PC}$ as a propositional proof system, we have to fix a reduction of **TAUT** to the language of all sets of algebraic formulas having no common roots. We could make this reduction from the reduction of formulas in CNF to sets of polynomials described in Section 1. However, the following (still standard) reduction is more natural.

There is still one variable for every Boolean variable (informally, `true` corresponds to 0, and `false` corresponds to 1). Our set of algebraic formulas will contain the formula $x^2 - x$ for each variable x occurring in the input tautology Θ and *one* additional formula $\varphi(\neg\Theta)$ (cf. *several* polynomials in the reduction described in Section 1). We define φ inductively: $\varphi(\neg\Phi) = (1 - \varphi(\Phi))$, and $\varphi(\Phi \supset \Psi) = (1 - \varphi(\Phi)) \cdot \varphi(\Psi)$ for Boolean formulas Φ and Ψ (one can easily extend φ to other logical connectives). We will refer to the algebraic formulas from the image of φ as *Boolean polynomials* (we will later show that for any of them we can derive $P^2 - P$). Sometimes we will also call Boolean any other polynomial P for which we can derive $P^2 - P$.

In what follows, we use $-P$ instead of $-1 \cdot P$, use other common mathematical notation, and omit straightforward calculations, for example,

$$\dfrac{\dfrac{\dfrac{\dfrac{-P + P}{-P + 1 \cdot P}}{(-1 + 1) \cdot P}}{0 \cdot P}}{0}.$$

Note that in $\mathcal{F}$-$\mathcal{PC}$ we can (and will in this paper):

1. Derive something from zero polynomial, because zero polynomial is trivially derivable from any other polynomial.

2. Omit some of the brackets and ignore the order of operands, because associativity and commutativity make it easy to derive similar formulas from each other.

3. Treat a polynomial $F - G$ as an equality $F = G$ and substitute G for an occurrence of F in any formula R containing F. This can be performed by extracting the multiplier $M = M(R)$ of this occurrence (define $M((P_0 + P_1)) = M(P_i)$ and $M((P_0 \cdot P_1)) = P_{1-i} \cdot M(P_i)$, where P_i is the part of the formula $(P_0 + P_1)$ or $(P_0 \cdot P_1)$ containing this occurrence of F; $M(P) = 0$ if P does not contain this occurrence; $M(F) = 1$ where F refers to the occurrence we mean), adding $(G - F) \cdot M$ to R and repeated carrying $G - F$ in brackets. In particular, we can substitute x for x^2 for any variable x.

4. Multiply equalities $F_1 = G_1$ and $F_2 = G_2$: Multiply $F_1 - G_1$ by F_2 and $F_2 - G_2$ by G_1; the sum of the obtained polynomials is $F_1F_2 - G_1G_2$, i.e., the equality $F_1F_2 = G_1G_2$.

5. Verify in the following simple way that F is derivable from G: open (some of the) brackets in both F and G, make appropriate substitutions using already derived equalities, group similar summands and compare the results. Clearly, one should care about the size of the proof obtained by opening the brackets.

[2]We could write "an $\mathcal{F}$-$\mathcal{PC}$ *refutation*", but write "*proof*" to emphasize that $\mathcal{F}$-$\mathcal{PC}$ is a Cook–Reckhow proof system. To reach a compromise between English and mathematics, the best way is to say that what we consider is a proof *of the fact that* $\{F_1, \ldots, F_m\}$ *have no common roots.*

2.2 $\mathcal{F}$-$\mathcal{NS}$

An $\mathcal{F}$-$\mathcal{NS}$ proof of a set $\{F_1, \ldots, F_m\}$ of algebraic formulas consists of two parts:

1. A set $\{G_1, \ldots, G_m\}$ of algebraic formulas.
2. An $\mathcal{F}$-$\mathcal{PC}$ derivation of 1 from the polynomial

$$\sum_{i=0}^{m} F_i G_i$$

without the use of the two main rules (2.1) (i.e., we use only the primitive rules).

2.3 Constant-depth $\mathcal{F}$-$\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{NS}$

When we refer to *constant-depth version* of either $\mathcal{F}$-$\mathcal{PC}$ or $\mathcal{F}$-$\mathcal{NS}$, we mean that the (initial and intermediate) polynomials are represented as formulas (terms) of depth bounded by a constant while the multiplication and the addition have arbitrary arity (i.e., we omit unnecessary brackets). The primitive rules must be modified accordingly.

The slight discrepancy with a trivial definition is motivated by the analogy between constant-depth $\mathcal{F}$-$\mathcal{PC}$ and constant-depth Frege systems. Depth-k $\mathcal{F}$-$\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{NS}$ are Cook–Reckhow proof systems for the language of all insolvable systems of depth-$(k-1)$ algebraic formulas (we consider $k \geq 3$ to capture, at least, sums of monomials and their formal products). We need to decrease here the depth by one to make the system complete (the primitive rules can turn every depth-$(k-1)$ formula into a sum of monomials using intermediate formulas of depth at most k).

Since constant-depth Boolean formulas are usually considered in the basis of $\neg$ and unbounded-arity $\vee$ (and sometimes $\wedge$ which is a shorthand: $\bigwedge_i x_i = \neg \bigvee_i \neg x_i$), we slightly modify the translation φ of Boolean formulas into algebraic formulas:

$$\varphi\left(\bigvee_{i=1}^{m} \Phi_i\right) = \prod_{i=1}^{m} \varphi(\Phi_i).$$

To get a *propositional* proof system, we must combine this reduction with a transformation of unbounded-depth Boolean formulas into constant-depth ones. Since we consider this system only in connection to constant-depth Frege systems (with modular counting), we do not need to fix this transformation.

Note that, *formally*, our reduction to systems of depth-$(k-1)$ algebraic formulas works well only for $k \geq 7$, because otherwise we are unable to translate even a formula in 3-CNF (in fact, we still get a complete proof system for a co-**NP**-hard problem, but the reduction must be further modified).

2.4 $\mathcal{F}$-$\mathcal{PC}>$ and $\mathcal{F}$-$\mathcal{NS}>$

In this subsection we discuss extensions of $\mathcal{F}$-$\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{NS}$ over $\mathbb{Q}$ by means of inequalities.

In the new systems $\mathcal{F}$-$\mathcal{PC}>$ and $\mathcal{F}$-$\mathcal{NS}>$ we work with equalities $P = 0$ instead of polynomials P. Therefore, Boolean formulas are translated into such equalities.

In $\mathcal{F}$-$\mathcal{PC}>$, we allow to replace an equality $P = 0$ by the two inequalities $P \geq 0$ and $-P \geq 0$, and to replace such pair of inequalities by the corresponding equality. We can still sum the equalities and multiply them by an arbitrary algebraic formula (as in (2.1)). The primitive rules (2.2)–(2.7)

can be applied to any subterm of an equality or of an inequality. The rules for working with inequalities are

$$\frac{P_1 \geq 0; \quad P_2 \geq 0}{(P_1 + P_2) \geq 0} \quad \text{and} \quad \frac{P_1 \geq 0; \quad P_2 \geq 0}{(P_1 \cdot P_2) \geq 0}.$$

There is also an axiom scheme $Q^2 \geq 0$ allowing to introduce the square of any algebraic formula Q. The goal is to derive the equality $1 = 0$.

Note that imposing a requirement that each intermediate polynomial has degree at most two gives the Lovasz-Schrijver calculus [LS91] (see also [Pud99]).

A proof of a set $\{F_1, \ldots, F_m\}$ of algebraic formulas in $\mathcal{F}$-$\mathcal{NS}>$ consists of:

1. A set $\{\mathcal{V}_l\}_{l=1}^{s}$ of subsets $\mathcal{V}_l$ of $[1..n]$.
2. Sets $\{G_i\}_{i=1}^{m}$ and $\{P_{lj}\}_{l\in[1..s],j\in[1..k]}$ of algebraic formulas.
3. An $\mathcal{F}$-$\mathcal{PC}$ derivation of 1 from the formula

$$\sum_{i=1}^{m} F_i G_i - \sum_{l=1}^{s} \left(\left(\prod_{v\in\mathcal{V}_l} F_v \right) \left(\sum_{j=1}^{k} P_{lj}^2 \right) \right)$$

using only the primitive rules, where the empty product means 1.

In fact, one derives in $\mathcal{F}$-$\mathcal{PC}>$ and $\mathcal{F}$-$\mathcal{NS}>$ elements of the cone generated by F_1, ..., F_m, $-F_1$, ..., $-F_m$ [BCR98]. Moreover, one could start with an input system of arbitrary inequalities $F_1 \geq 0$,...,$F_m \geq 0$ (not necessarily including inequalities $-F_i \geq 0$), then the completeness of the corresponding proof systems follows from Positivstellensatz [BCR98] (note that to make a *new propositional* proof system one also has to update φ).

Alternatively, one might consider weaker systems in which one does not convert equalities into pairs of inequalities, but rather derives from $\{F_i\}_{i=1}^{m}$ an element F from the ideal generated by $\{F_i\}_{i=1}^{m}$ (in $\mathcal{F}$-$\mathcal{PC}$ or $\mathcal{F}$-$\mathcal{NS}$, respectively). We say that there is a proof in the corresponding weaker system, if there are algebraic formulas H_j such that $\sum_j H_j^2 + F = -1$. Note that lower bounds on the degree of proofs for this kind of systems were established in [Gri99, Gri00, GV01]. The completeness of these weaker systems also follows from Positivstellensatz.

2.5 $\mathcal{F}$-$\mathcal{PC}\sqrt{}$

The (*radical*) system $\mathcal{F}$-$\mathcal{PC}\sqrt{}$ is the system $\mathcal{F}$-$\mathcal{PC}$ extended by the rule

$$\frac{P^2}{P}.$$

One can also define the system $\mathcal{PC}\sqrt{}$ similarly, and the system $\mathcal{F}$-$\mathcal{NS}\sqrt{}$ by including the radical rule in the list of primitive rules. Note that this rule is in accordance with Nullstellensatz [vdW31] since P vanishes on the variety given by the equalities $F_1 = \ldots = F_m = 0$ if and only if a certain power P^d belongs to the ideal generated by $F_1, \ldots, F_m$. Although this rule looks redundant because every Boolean (i.e., comprising polynomials $x_i^2 - x_i$ for $1 \leq i \leq n$) ideal is radical, this rule apparently could accelerate proofs in some cases.

Consider, for example, the following issue. We have defined an $\mathcal{F}$-$\mathcal{PC}$ (resp., $\mathcal{F}$-$\mathcal{NS}$, $\mathcal{PC}$, etc.) proof of Φ as a derivation of 1 from the set $S = \{\varphi(\neg\Phi), x_1^2 - x_1, \ldots, x_n^2 - x_n\}$. Frequently, such a derivation is called "*refutation*" instead of "proof". There is another possibility to prove

that Φ is a tautology: let us call a "proper proof" a derivation of $\varphi(\Phi)$ from $\{x_1^2 - x_1, \ldots, x_n^2 - x_n\}$. Formally, "refutations" and "proper proofs" give different propositional proof systems. (Note that in customary $\mathcal{PC}$ and $\mathcal{NS}$ using "proper proofs" instead of "refutations" needs updating the translation of formulas into polynomials since we defined it only for CNFs; for many Boolean formulas Φ, this results in exponentially large polynomials).

Trivially, any proof in a "proper proof" system can be transformed (with a negligible increase in size) into a proof in a "refutation" system. Can we do it in the reverse direction? For Boolean polynomial $F = \varphi(\Phi)$, one can multiply every line of a "refutation" of $(1 - F)$ by F, obtaining a "proper proof" of F. Note that the axiom $(1 - F)$ transforms into $F(1 - F)$; we will later see (Lemma 1) that for any F from the image of φ the formula $F(1 - F)$ is easily derivable in $\mathcal{F}$-$\mathcal{PC}$; it is also easily derivable in $\mathcal{PC}$, because it is an element of the ideal generated by $x_i^2 - x_i$, i.e., $\sum H_i(x_i^2 - x_i)$.

However, for non-Boolean F this may not work. Although it is impossible to transform a "refutation" of $(1 - F)$ into a "proper proof" of F, we now show how to transform a "refutation" of $(1 - zF)$ (where z is a new variable) into a "proper proof" of F, using the radical rule. For this, we apply "the trick of Rabinowitsch" [vdW31, Part 2, Chapter 11].

A "proper proof" of F in $\mathcal{NS}\sqrt{}$ is a "proper proof" of F in $\mathcal{NS}$ followed by several applications of the radical rule to $\sum_i F_i G_i$. Consider an $\mathcal{NS}\sqrt{}$ "refutation" (this is the same as $\mathcal{NS}$ "refutation") $(1 - zF)G + \sum_i F_i G_i = 1$. Substituting $z = 1/F$ in this equality and cleaning the denominator, we get $F^d = \sum_i F_i G_i'$, moreover, $d \le \max_i \deg(G_i)$, $\deg(G_j') \le \deg(F^d) + \max_i \deg(G_i) \le (\deg(F) + 1) \cdot \max_i \deg(G_i)$. Now applying several times the radical rule this provides an $\mathcal{NS}\sqrt{}$ derivation of F from $\{F_1, \ldots, F_m\}$ of degree growing at most polynomially in the degree of a $\mathcal{NS}\sqrt{}$ derivation of 1 from $\{1 - zF, F_1, \ldots, F_m\}$

To get a "proper proof" of F in $\mathcal{PC}\sqrt{}$ from a "refutation" of $(1-zF)$ in $\mathcal{PC}\sqrt{}$, we verify by induction along the proof that for each its line $\sum F_i G_i$, one can derive a polynomial $F^{d_z+1} \sum F_i G_i|_{z=1/F}$ by few applications of the rules of $\mathcal{PC}\sqrt{}$, where d_z is the maximum degree of z in the intermediate polynomials of the "refutation" (including the polynomials Q used in the $\dfrac{P}{P \cdot Q}$ rule).

Suppose that $\sum F_i G_i$ is obtained as the sum of two already derived polynomials $\sum F_i G_{1i}$ and $\sum F_i G_{2i}$. By induction one can derive $F^{d_z+1} \sum F_i G_{1i}|_{z=1/F} + F^{d_z+1} \sum F_i G_{2i}|_{z=1/F}$. Clearly, this gives $F^{d_z+1} \sum F_i G_i|_{z=1/F}$. If $\sum F_i G_i$ is obtained as the product of an already derived polynomial $\sum F_i G_{i1}$ and a polynomial Q, by induction we can derive $F^{2d_z+1} \sum F_i G_i$ (we multiply by $F^{d_z} Q|_{z=1/F}$ instead of Q). Multiplying by $F \sum F_i G_i$ and using the radical rule, one gets $F^{d_z+1} \sum F_i G_i$. Finally, we get "a proper $\mathcal{PC}\sqrt{}$ proof" of F^{d_z+1}. Then the multiplication by F^c (where $c + d_z + 1$ is the nearest power of two) and the repeated application of the radical rule allows one to derive F itself.

The same arguments can be conducted for $\mathcal{F}$-$\mathcal{NS}\sqrt{}$. We leave the corresponding question for $\mathcal{F}$-$\mathcal{PC}\sqrt{}$ open. (Note that in the cases of $\mathcal{F}$-$\mathcal{NS}$ and $\mathcal{F}$-$\mathcal{PC}$ degree bounds do not suffice for obtaining bounds on the size of derivation.)

3 $\mathcal{F}$-$\mathcal{PC}$ simulates Frege systems

In this section it does not matter whether we consider $\mathcal{F}$-$\mathcal{PC}$ over $\mathbb{Q}$ or over $\mathbb{Z}_p$: we use only the existence of the constants 0, 1 and -1.

Theorem 1. *The system $\mathcal{F}$-$\mathcal{PC}$ polynomially simulates Frege systems.*

Proof. We consider Hilbert's system: The axioms are

(A1) $\Gamma \supset (\Delta \supset \Gamma)$,

(A2) $(\neg\Delta \supset \neg\Gamma) \supset ((\neg\Delta \supset \Gamma) \supset \Delta)$,

(A3) $(\Gamma \supset (\Delta \supset \Lambda)) \supset ((\Gamma \supset \Delta) \supset (\Gamma \supset \Lambda))$.

The only rule of the inference is modus ponens: Γ and $\Gamma \supset \Delta$ imply Δ.

The main part of the proof is the following lemma.

Lemma 1. *For any Boolean formula Φ, the shortest $\mathcal{F}$-$\mathcal{PC}$ proof of the polynomial $(\varphi(\Phi))^2 - \varphi(\Phi)$ has size polynomial in size of Φ.*

Proof. We prove it by induction on the construction of Φ. Let $G = \varphi(\Gamma)$, $D = \varphi(\Delta)$, and suppose that $G^2 - G$ and $D^2 - D$ have derivations of sizes $\gamma \leq c|\Gamma|^2$ and $\delta \leq c|\Delta|^2$ respectively (the constant c will be clear from what follows). Note that using the primitive rules we can easily derive $R = (1-G)^2 - (1-G)$ from $G^2 - G$. Hence, in the case $\Phi = \neg\Gamma$ the polynomial $(\varphi(\Phi))^2 - \varphi(\Phi)$ has a derivation of size at most $c|\Phi|^2$ (if c is large enough). We now consider the case $\Phi = (\Gamma \supset \Delta)$. By multiplying $D^2 - D$ by $(1-G)^2$, we obtain $P = (1-G)^2D^2 - (1-G)^2D$. On the other hand, we can derive $Q = (1-G)^2D - (1-G)D$ by multiplying R by D. Summing P and Q, we derive $((1-G)D)^2 - (1-G)D$ which is $(\varphi(\Phi))^2 - \varphi(\Phi)$. The size of this proof (from $G^2 - G$ and $D^2 - D$) is upper bounded by $d|\Phi|$ for some constant d, i.e., the size of the proof from axioms is at most $d|\Phi| + c|\Gamma|^2 + c|\Delta|^2 = c(|\Gamma| + |\Delta|)^2 + d|\Phi| - 2c|\Gamma||\Delta|$. Now, choose $c \geq 2d$; then the size of our proof is at most $c|\Phi|^2$. □

We now show that φ translates axioms into polynomials having $\mathcal{F}$-$\mathcal{PC}$ derivations of size polynomial in $|\Gamma|$, $|\Delta|$ and $|\Lambda|$.

(A1) Let $G = \varphi(\Gamma)$, $D = \varphi(\Delta)$. Since we can derive $G^2 - G$, by commutativity and distributivity we have $(1-G)G$. Then we multiply it by $(1-D)$ and by commutativity and associativity we obtain $(1-G)((1-D)G) = \varphi(\Gamma \supset (\Delta \supset \Gamma))$.

(A2) Here, we need to derive $(1 - D(1-G)) \cdot ((1-DG)D)$ up to simplifications. Opening the brackets gives $(D - D^2) + (GD^2 - D^2G) + (D^3G - G^2D^3)$; it now remains to derive this formula. The first two summands can be easily derived from $D^2 - D$ and from the axioms, respectively. The third summand is $G^2 - G$ multiplied by $-D^3$.

(A3) Here, we need to derive $(1-(1-G)((1-D)L)) \cdot ((1-(1-G)D)((1-G)L))$, where $L = \varphi(\Lambda)$. This formula is equal to

$$(G^3L^2 - 3G^2L^2 + 3GL^2 - L^2)(D^2 - D) + (G^2D - 2GD + G + D - 1)(L^2 - L) - L^2(G^2 - G)$$

(it can be verified by opening all brackets), which is a sum of $G^2 - G$, $D^2 - D$ and $L^2 - L$ multiplied by appropriate polynomials.

It is clear that modus ponens can be proved using size linear in the sizes of the formulas Γ and $\Gamma \supset \Delta$: having $\varphi(\Gamma) = G$ and $\varphi(\Gamma \supset \Delta) = (1-G)D$, we can multiply G by D, add the result to $(1-G)D$, and after simplifications we have D, i.e., $\varphi(\Delta)$. Therefore, we can indeed transform the proof of a Boolean formula Θ in Hilbert's system into a derivation of $\varphi(\Theta)$ in $\mathcal{F}$-$\mathcal{PC}$ from the polynomials $x_i^2 - x_i$ with only a polynomial increase in size (moreover, the only point where non-linear increase can occur is the translation of the axioms of Hilbert's system). Summing $\varphi(\Theta)$ with the axiom $\varphi(\neg\Theta)$, we get 1. □

4 $\mathcal{F}$-$\mathcal{NS}$ is just tree-like $\mathcal{F}$-$\mathcal{PC}$

How much of the power of $\mathcal{F}$-$\mathcal{PC}$ is taken away by replacing it with $\mathcal{F}$-$\mathcal{NS}$? Although we do not know whether $\mathcal{F}$-$\mathcal{NS}$ can polynomially simulate $\mathcal{F}$-$\mathcal{PC}$, in this section we show that $\mathcal{F}$-$\mathcal{NS}$ still can polynomially simulate Frege systems. Namely, we show that $\mathcal{F}$-$\mathcal{NS}$ can polynomially simulate tree-like $\mathcal{F}$-$\mathcal{PC}$ ("tree-like" means that every derived polynomial is used only once; if we need it again, we must derive it once more), cf. [BIK+97] which proves that $\mathcal{NS}$ polynomially simulates tree-like $\mathcal{PC}$ over $\mathbb{Z}_p$. The claim now follows since tree-like Frege systems have the same power as usual (DAG-like) ones (see, e.g., [Kra95]), and the proof of Theorem 1 translates tree-like Frege proofs into tree-like $\mathcal{F}$-$\mathcal{PC}$ proofs.

Clearly, tree-like $\mathcal{F}$-$\mathcal{PC}$ simulates $\mathcal{F}$-$\mathcal{NS}$, cf. the fact that tree-like $\mathcal{PC}$ simulates $\mathcal{NS}$ [BIK+97]. Therefore, the following theorem establishes the equivalence between tree-like $\mathcal{F}$-$\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{NS}$.

Theorem 2. *$\mathcal{F}$-$\mathcal{NS}$ polynomially simulates tree-like $\mathcal{F}$-$\mathcal{PC}$.*

Proof. We first show that a tree-like $\mathcal{F}$-$\mathcal{PC}$ proof can be transformed (with at most polynomial increase in size) into two derivations:

1. A derivation π_T of some formula T from the axioms using no primitive rules.
2. A derivation $\pi_{T,1}$ of 1 from T using only primitive rules.

We transform it inductively; at each intermediate step of our induction we will have two derivations: a "normal" derivation π_U of some formula U from the axioms, and a derivation $\pi_{U,1}$ of 1 from U using only the primitive rules.

We move the applications of the primitive rules from the first derivation to the second derivation one by one. Consider the last application of a primitive rule in π_U. Let S be a subformula to which this rule is applied. Note that if we omit this application, S will remain as a whole till the end of the proof, and this will be the only difference between the old final formula U of the proof and the new final formula U'. Let us apply the same rule to U'; we then obtain U and therefore have a derivation $\pi_{U',1}$ deriving 1 from U'. Observe that every step of this induction increases the proof size at most by the square of the size of the original proof.

Note that π_T is just the syntactic tree of the term T. We define the coefficients G_i of our $\mathcal{F}$-$\mathcal{NS}$ proof $\sum_{i=1}^m F_i G_i$ (see Subsection 2.2 above) inductively. The argument of $M_i(\cdot)$ is a subtree of π_T. For the axioms F_i, the formula $M_i(F_i)$ is defined to be one if $i = j$, and zero otherwise. If the root R of a subtree π_R is derived as the sum of two formulas P and Q, we define $M_i(\pi_R) = (M_i(\pi_P) + M_i(\pi_Q))$, where π_P and π_Q are the subtrees corresponding to the proofs of P and Q respectively. If the root of π_R is derived as $(P \cdot Q)$, where P is an already derived formula, then $M_i(\pi_R) = (M_i(\pi_P) \cdot Q)$. Finally, we let the coefficient G_i be $M_i(\pi_T)$.

Clearly, the size of every G_i is less or equal to the size of T. We must now present the proof of 1 from $\sum_{i=1}^m F_i G_i$ using only the primitive rules.

If suffices to show that there is a derivation (of size polynomial in the size of T) of T from this sum using only the primitive rules. The proof is by induction on the construction of M_i's. Suppose that we have derived the term T in which some subterms R_j are replaced by

$$\left(\sum_{i=1}^{m} F_i M_i(\pi_{R_j}) \right). \tag{4.1}$$

Then we rearrange one of such sums $\sum_i F_i M_i(\pi_R)$. If $M_i(\pi_R)$'s were obtained as $(M_i(\pi_P) \cdot Q)$, we transform sum (4.1) into $((\sum_i F_i M_i(\pi_P)) \cdot Q)$. If $M_i(\pi_R)$'s were obtained as $(M_i(\pi_P) + M_i(\pi_Q))$,

then we transform (4.1) into $((\sum_i F_i M_i(\pi_P)) + (\sum_j F_j M_j(\pi_Q)))$. If the argument of M_i's is an axiom F_j, then we simplify (4.1) to F_j. □

Corollary 1. *$\mathcal{F}$-$\mathcal{NS}$ polynomially simulates Frege systems.*

Proof. Note that tree-like Frege systems polynomially simulate Frege systems [Kra95], and the proof of Theorem 1 translates tree-like Frege proofs into tree-like $\mathcal{F}$-$\mathcal{PC}$ proofs. □

5 Constant-depth $\mathcal{F}$-$\mathcal{PC}$ over $\mathbb{Z}_p$ simulates constant-depth Frege systems with MOD_p gates

In this section we show that constant-depth $\mathcal{F}$-$\mathcal{PC}$ over $\mathbb{Z}_p$ polynomially simulates constant-depth Frege systems with MOD_p gates. (The depth of algebraic formulas may be bounded by a different constant than the depth of Boolean formulas).

Since constant-depth Boolean formulas are usually considered in the basis of $\neg$ and unbounded-arity $\vee$ (and sometimes $\wedge$ which is a shorthand: $\bigwedge_i x_i = \neg \bigvee_i \neg x_i$), we switch to this basis for this section. Note that the axioms of Hilbert's system translate into the same polynomials in $\varphi(\Gamma)$, $\varphi(\Delta)$, $\varphi(\Lambda)$ as before, and so does modus ponens. It is easy to see that despite we modified φ for the constant-depth version of $\mathcal{F}$-$\mathcal{PC}$, still it translates Boolean formulas into Boolean polynomials, i.e., the analog of Lemma 1 still holds. We summarize that the proof of Theorem 1 still works for constant-depth Boolean formulas, and transforms constant-depth Frege proofs into constant-depth $\mathcal{F}$-$\mathcal{PC}$ proofs.

A Frege system with MOD_p gates (see, e.g., [Kra95]) includes propositional connectives $\mathrm{MOD}_{p,i}$ of unbounded arity ($0 \leq i \leq p-1$). Informally, $\mathrm{MOD}_{p,i}(x_1, \ldots, x_k)$ means that the number of x_i's having the value `true` equals i modulo p. We add the axiom schemes

$$\mathrm{MOD}_{p,0}(\emptyset)$$

and

$$\neg\mathrm{MOD}_{p,i}(\emptyset)$$

for each $i = 1, \ldots, p-1$. For each nonnegative integer k, we add also the axiom schemes

$$\begin{aligned}&\mathrm{MOD}_{p,i}(\Phi_1, \ldots, \Phi_k, \Phi_{k+1}) \equiv \\ &\equiv ((\mathrm{MOD}_{p,i}(\Phi_1, \ldots, \Phi_k) \wedge \neg\Phi_{k+1}) \vee (\mathrm{MOD}_{p,(i-1) \bmod p}(\Phi_1, \ldots, \Phi_k) \wedge \Phi_{k+1}).\end{aligned}$$

(Here $\equiv$ and $\wedge$ are just shorthands). To translate formulas with MOD_p connectives into algebraic formulas over $\mathbb{Z}_p$ we extend φ to MOD_p gates by

$$\varphi(\mathrm{MOD}_{p,i}(\Phi_1, \ldots, \Phi_k)) = (k - i - \varphi(\Phi_1) - \ldots - \varphi(\Phi_k))^{p-1}.$$

To see that the obtained formula is Boolean, it is sufficient to prove $A^p - A$ where A denotes $k - i - \varphi(\Phi_1) - \ldots - \varphi(\Phi_k)$ (then $A^{2p-2} - A^{p-1}$ follows easily). Note that we can represent A as the sum of Boolean polynomials $F_1, \ldots, F_{k+((p-i) \bmod p)}$, where $F_j = 1 - \varphi(\Phi_j)$ for $1 \leq j \leq k$ and $F_j = 1$ otherwise. When we open brackets in $(\sum_j F_j)^p$ and group similar "monomials", all summands except F_j^p cancel because p divides their coefficients. Since F_j are Boolean, the claim follows, i.e., the analog of Lemma 1 holds even for constant-depth Boolean formulas with MOD_p gates. Namely, for such formula Φ, the algebraic formula $(\varphi(\Phi))^2 - \varphi(\Phi)$ has polynomial-size constant-depth $\mathcal{F}$-$\mathcal{PC}$ proof over $\mathbb{Z}_p$.

Theorem 3. *Constant-depth $\mathcal{F}$-$\mathcal{PC}$ over $\mathbb{Z}_p$ polynomially simulates constant-depth Frege systems with MOD_p gates.*

Proof. By the above discussion concerning the proofs of Lemma 1 and Theorem 1, it suffices to show that the axiom schemes for $\mathrm{MOD}_{p,i}$ connectives have short proofs in $\mathcal{F}$-$\mathcal{PC}$ over $\mathbb{Z}_p$.

The schemes for $\emptyset$ translate into trivial formulas involving no variables. The only non-trivial case is that of

$$1-(1-(1-B)S)\cdot(1-(1-S)B), \tag{5.1}$$

where

$$\begin{aligned}
&B=(A+F)^{p-1},\\
&S=(1-(1-A^{p-1})(1-F))(1-(1-(A+1)^{p-1})F),\\
&A=k-i-\varphi(\Phi_1)-\ldots-\varphi(\Phi_k),\\
&\text{and } F=1-\varphi(\Phi_{k+1}).
\end{aligned}$$

Note that (5.1) can be transformed into $(B-S)^2$ (to verify, open all brackets and use B^2-B and S^2-S). Therefore, it suffices to prove $B-S$.

The formula B can be transformed as follows:

$$(A+F)^{p-1}=A^{p-1}+F\sum_{j=0}^{p-2}\binom{p-1}{j}A^j=A^{p-1}+F((A+1)^{p-1}-A^{p-1}).$$

On the other hand, opening the external brackets in S and using F^2-F gives

$$1-(1-A^{p-1})(1-F)-(1-(A+1)^{p-1})F=A^{p-1}+F((A+1)^{p-1}-A^{p-1}).$$

□

6 PHP

In this section we present a short proof of the propositional pigeon-hole principle in constant-depth $\mathcal{F}$-$\mathcal{PC}$ over $\mathbb{Q}$, moreover, this proof can be conducted in constant-depth $\mathcal{F}$-$\mathcal{NS}$.

PHP_n is usually formulated as

$$\left(\bigwedge_{i=1}^{n+1}\bigvee_{k=1}^{n}p_{ik}\right)\supset\bigvee_{k=1}^{n}\bigvee_{1\le i<j\le n+1}(p_{ik}\wedge p_{jk}),$$

and its negation written in the basis $\{\neg,\vee\}$ is

$$\neg\left(\left(\bigvee_{i=1}^{n+1}\neg\bigvee_{k=1}^{n}p_{ik}\right)\vee\bigvee_{k=1}^{n}\bigvee_{1\le i<j\le n+1}\neg(\neg p_{ik}\vee\neg p_{jk})\right).$$

We now give a short $\mathcal{F}$-$\mathcal{PC}$ proof of the $\mathcal{F}$-$\mathcal{PC}$ version of $\neg\mathrm{PHP}_n$:

$$1-\prod_{i=1}^{n+1}(1-p_{i1}p_{i2}\ldots p_{in})\ \cdot\ \prod_{1\le i<j\le n+1}\prod_{k=0}^{n}(1-(1-p_{ik})(1-p_{jk})) \tag{6.1}$$

or, equivalently,

$$\forall i, \qquad p_{i1}p_{i2}\ldots p_{in} \tag{6.2}$$

$$\forall i < j, \forall k, \qquad (1-p_{ik})(1-p_{jk}). \tag{6.3}$$

We start by (informally) switching to another set of variables

$$q_{ik} = (1-p_{ik}) \prod_{1\le l<k} p_{il}.$$

In the real derivation, these variables are replaced by the corresponding formulas. We can easily prove

$$\forall i, \qquad 1 - q_{i1} - q_{i2} - \ldots - q_{in} \tag{6.4}$$

$$\forall i, \forall j < i, \forall k, \qquad q_{ik}q_{jk} \tag{6.5}$$

$$\forall i, \forall k, \forall l < k, \qquad q_{ik}q_{il} \tag{6.6}$$

$$\forall i, k, \qquad q_{ik}^2 - q_{ik}. \tag{6.7}$$

Indeed, all summands in (6.4) after opening brackets cancel except (6.2). Then, (6.5) is just (6.3) multiplied by $\prod_{1\le l<k}(p_{il}p_{jl})$. The formula (6.6) is a product containing $p_{il}(1-p_{il})$. Finally, (6.7) follows from

$$q_{ik} = \varphi\left(\neg p_{ik} \vee \bigvee_{1\le l<k} p_{il}\right)$$

(see Lemma 1).

We sum (6.4) for all i's and rearrange it as

$$(n+1) - \sum_{k=1}^{n} x_k, \tag{6.8}$$

where $x_k = q_{1k} + \ldots + q_{n+1,k}$. Note that x_k is Boolean: open the brackets in

$$(q_{1k} + \ldots + q_{n+1,k})^2 - (q_{1k} + \ldots + q_{n+1,k})$$

and use (6.5) and (6.7).

To derive 1 from (6.8) (which is an instance of the subset sum problem [IPS99]), we inductively derive the polynomial

$$S_n(S_n - 1)\ldots(S_n - n), \tag{6.9}$$

where $S_i = x_1 + x_2 + \ldots + x_i$.

We start from $S_1(S_1-1)$ which is simply $x_1^2 - x_1$. The induction step is to prove

$$S_i(S_i - 1)\ldots(S_i - i) \tag{6.10}$$

from

$$S_{i-1}(S_{i-1} - 1)\ldots(S_{i-1} - (i-1)). \tag{6.11}$$

This derivation itself will be done inductively too. To stress the difference between the two induction arguments, we denote S_{i-1} by S and x_i by x. Multiply (6.11) by $S + (i+1)x - i$. Opening brackets in the last two terms $(S-(i-1))(S+(i+1)x-i)$ of the product, adding $-i(x^2-x)$ and bracketing

again, we get $(S+ix-(i-1))(S+x-i)$ (to verify, open the brackets and notice that the difference is $i(x^2-x)$). We now turn our attention to the previous expression $(S-(i-2))$ in brackets, open the brackets in $(S-(i-2))(S+(i-1)x-(i-2))$ etc. Finally, we arrive at (6.10).

To derive 1 from (6.9), substitute in it $n+1$ for S_n (use (6.8)) and multiply the result by $1/(n+1)!$.

Observe that this proof can be made tree-like and hence may be conducted in $\mathcal{F}$-$\mathcal{NS}$. On the other hand, observe that all the involved formulas in the proof are of depth bounded by a constant.

Remark. Note that the above version of PHP is given by polynomials of large degree (see (6.1)). On the other hand, in [Raz98] (see also [IPS99]) the injective PHP given by polynomials of degree at most two was studied. These are essentially the polynomials (6.4)–(6.7), where q_{ik} are treated as variables and not as shorthands for formulas. The paper [Raz98] establishes the lower bound $\Omega(n)$ on the degree of $\mathcal{PC}$ proofs of this set of polynomials. This implies an exponential lower bound [IPS99, Theorem 6.7] on the size of the shortest $\mathcal{PC}$ proof of this set of polynomials. Our short derivation of 1 from (6.4)–(6.7) demonstrates an exponential separation between $\mathcal{PC}$ and constant-depth $\mathcal{F}$-$\mathcal{PC}$ (and also $\mathcal{F}$-$\mathcal{NS}$) as proof systems for the language of all insolvable systems of polynomial equations. However, it does not give a separation of these systems as *propositional* proof systems, because the formulation of PHP studied in [Raz98] may be not in the image of $\mathcal{PC}$'s translation of Boolean formulas. The separation between propositional proof systems $\mathcal{PC}$ and constant-depth $\mathcal{F}$-$\mathcal{PC}$ (and $\mathcal{F}$-$\mathcal{NS}$) is given in Section 7 by means of Tseitin's tautologies.

By the same token, the subset sum problem provides the same separation result as the formulation of PHP from [Raz98] does, because [IPS99] gives an exponential lower bound on the size of the shortest $\mathcal{PC}$ proof of any instance of the subset sum problem, and we obtained a short constant-depth $\mathcal{F}$-$\mathcal{PC}$ (and even $\mathcal{F}$-$\mathcal{NS}$) proof of its instance (6.8) (in the Boolean variables x_i).

7 Tseitin's tautologies

In this section we show an exponential gap between lengths of proofs in $\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{PC}$ viewed as *propositional* proof systems. First, we show that Tseitin's tautologies have short $\mathcal{F}$-$\mathcal{PC}$ proofs while they have no $\mathcal{PC}$ proofs over any field of characteristic different from two. Afterwards, using the generalization of Tseitin's tautologies given in [BGIP99], we show how to handle the remaining case; in fact, the generalization works for any field containing a p-th root of unity for some prime p. Our $\mathcal{F}$-$\mathcal{PC}$ proof can be conducted even in constant-depth $\mathcal{F}$-$\mathcal{NS}$. This exhibits an exponential separation between the *propositional* proof systems $\mathcal{PC}$ and constant-depth $\mathcal{F}$-$\mathcal{NS}$ over any field.

7.1 Fields of characteristic different from two

Let $\mathcal{G}=(V,E)$ by any undirected graph with an odd number of vertices and with expansion ε, i.e., for any subset $S\subseteq V$ of cardinality at most $|V|/2$, the graph $\mathcal{G}$ has at least $(1+\varepsilon)|S|$ neighbors of S. For any number of vertices, there are such $\mathcal{G}$'s of degree bounded by a constant c (see, e.g., [Alo86]).

Tseitin's tautology for $\mathcal{G}$ is given by the Boolean formula

$$\neg \bigwedge_{v\in V} \bigoplus_{e\in E_v} x_e \tag{7.1}$$

(where E_v is the set of edges incident to v) and its negation is the following formula in CNF:

$$\bigwedge_{v\in V} \bigwedge_{\substack{i_1,\ldots,i_{\deg(v)}\in\{0,1\},\\ i_1\oplus\ldots\oplus i_{\deg(v)}=0}} \left(\bigvee_{k:\, i_k=0} x_{e(v,i_k)} \vee \bigvee_{l:\, i_l=1} \neg x_{e(v,i_l)} \right), \tag{7.2}$$

where $e(v,i)$ denotes the i-th edge in E_v. There is one variable x_e for every edge e of $\mathcal{G}$.

The $\mathcal{PC}$ translation of (7.2) is given by the polynomials

$$x_e^2 - x_e, \tag{7.3}$$

$$\prod_{k:\, i_k=0} x_{e(v,i_k)} \cdot \prod_{l:\, i_l=1} (1 - x_{e(v,i_l)}) \tag{7.4}$$

where in (7.3) e ranges over E, in (7.4) v ranges over V and $(i_1,\ldots,i_{\deg(v)})$ ranges over $\{0,1\}^{\deg(v)}$ and $i_1 \oplus \ldots \oplus i_{\deg(v)} = 0$. (There are $\sum_{v\in V} 2^{\deg(v)-1}$ polynomials in (7.4), each of degree at most c).

In [BGIP99] a linear degree lower bound for another formulation of this problem is shown:

$$X_e^2 - 1, \tag{7.5}$$

$$1 + \prod_{e\in E_v} X_e, \tag{7.6}$$

where in (7.5) e ranges over E, and in (7.6) v ranges over V (note that there are no $X_e^2 - X_e$ polynomials). The following argument shows that this lower bound holds also for (7.3)–(7.4).

First, replace all occurrences of X_e's in (7.5)–(7.6) by $(2x_e - 1)$. We get

$$4(x_e^2 - x_e), \tag{7.7}$$

$$1 + \prod_{e\in E_v} (2x_e - 1). \tag{7.8}$$

Note that any low degree $\mathcal{PC}$ proof of (7.3)–(7.4) can be easily extended to a low degree proof of (7.7)–(7.8): the polynomials (7.3) and (7.7) can be obtained from each other by multiplying by four, and (7.4) and (7.8) (fix some v now) are two constant-degree polynomials that have the same values on $\{0,1\}^{|E_v|}$ and therefore differ by

$$\sum_{e\in E_v} (G_{v,e}(x_e^2 - x_e)),$$

where $G_{v,e}$ are some constant-degree polynomials.

Now consider any such low degree proof of (7.7)–(7.8) and replace in it all occurrences of x_e's by $(X_e+1)/2$. We obtain a low degree proof of (7.5)–(7.6). But such proofs do not exist [BGIP99]. Thus, our assumption that there is a low degree $\mathcal{PC}$ proof of (7.3)–(7.4) is false. Therefore, by [IPS99, Theorem 6.2] there are no polynomial-size proofs of Tseitin's tautologies (7.2) in $\mathcal{PC}$.

However, there are such proofs in $\mathcal{F}$-$\mathcal{PC}$. Consider the $\mathcal{F}$-$\mathcal{PC}$ translation of (7.2). The system (7.3)–(7.4) can be obtained from it very easily. We already mentioned that we can derive (7.7)–(7.8) from it. Now change in these algebraic formulas all occurrences of x_e's by $(X_e + 1)/2$, where X_e denotes the expression $(\frac{1}{2} \cdot (x_e + 1))$. We arrive at (7.5)–(7.6).

Consider the polynomials (7.6) as equalities $\prod_{e\in E_v} X_e = -1$ and multiply them one by one, substituting 1 for X_e^2's using (7.5) (cf. Subsection 2.1). Finally, we arrive at the equality $1 = (-1)^{|V|}$, i.e., to the polynomial 2. It remains to divide it by two. One can verify that our $\mathcal{F}$-$\mathcal{PC}$ proof can be conducted in constant-depth $\mathcal{F}$-$\mathcal{NS}$.

7.2 Fields of arbitrary characteristic

One could generalize this construction (see [BGIP99]) starting with a prime p and an expander $\mathcal{G} = (V, E)$ such that $|V| \equiv 1 \pmod p$. Considering $\mathcal{G}$ as a directed (in arbitrary way) graph we assign to each its edge e a variable X_e which satisfy the following conditions. If $\bar{e}$ is the edge with the orientation opposite to e, then we include the polynomial $X_e X_{\bar{e}} - 1$ in the input system of polynomials. We also include the polynomials $X_e^p - 1$ (which replace (7.5)). Finally, for each vertex v we include the polynomial $\prod X_e - \omega$, where ω is a p-th root of unity (we assume that the ground field contain ω) and the product ranges over all the edges e emanating from v (this polynomial replaces (7.6)). The obtained system can be represented by a Boolean formula (cf. (7.1)) which we denote by Ψ_p.

Similarly to above, one could produce a constant-depth proof of Ψ_p in $\mathcal{F}$-$\mathcal{NS}$. On the other hand, a linear lower bound on the degree of the shortest $\mathcal{PC}$ proof of Ψ_p over any field of characteristic distinct from p is established in [BGIP99].

8 Further research

Since $\mathcal{F}$-$\mathcal{NS}$ polynomially simulates Frege systems, proving lower bounds for it (and hence for $\mathcal{F}$-$\mathcal{PC}$) seems a hard problem. There is however a lot of apparently easier problems related to the constant-depth versions of $\mathcal{F}$-$\mathcal{PC}$ and $\mathcal{F}$-$\mathcal{NS}$. For example, all we know about constant-depth $\mathcal{F}$-$\mathcal{PC}$ over $\mathbb{Q}$ (resp., over $\mathbb{Z}_p$) is that

- it polynomially simulates constant-depth Frege systems (resp., with MOD_p gates);
- it polynomially simulates $\mathcal{PC}$ over $\mathbb{Q}$ (resp., over $\mathbb{Z}_p$);
- it has polynomial-size proofs of PHP (over $\mathbb{Q}$) and Tseitin's tautologies.

What kind of Frege systems (without extension rules) could simulate constant-depth $\mathcal{F}$-$\mathcal{PC}$? We know even less about constant-depth $\mathcal{F}$-$\mathcal{NS}$. Does it simulate tree-like constant-depth $\mathcal{F}$-$\mathcal{PC}$? Finally, we do not know any lower bounds even for constant-depth $\mathcal{F}$-$\mathcal{NS}$ over any field.

We have also introduced several extensions of $\mathcal{F}$-$\mathcal{PC}$. It would be interesting to clarify whether these extensions actually amplify $\mathcal{F}$-$\mathcal{PC}$. Observe that there is a degree two $\mathcal{PC}{>}$ (even $\mathcal{NS}{>}$) proof of the subset sum problem with positive weights $\sum a_i x_i = m$, where integers $a_i > 0$, $m < 0$ [Gri00]. Note that these integers are written in *binary* form. We ask whether there is a short proof of this problem in $\mathcal{F}$-$\mathcal{PC}$. Note that a linear lower bound $\Omega(n)$ on degrees of its $\mathcal{PC}$ proofs is shown in [IPS99]. This exhibits a gap between $\mathcal{PC}{>}$ and $\mathcal{PC}$ over $\mathbb{Q}$ (due to [IPS99, Theorem 6.2]).

Another extension of $\mathcal{F}$-$\mathcal{PC}$ (considered in [Pit97]) emerges when one allows to replace an algebraic formula with arbitary algebraic formula representing the same polynomial without verification of their equivalence. Such verification of course could be done in **BPP**. As an example for which it is not clear how to verify quickly a formula in a deterministic way (say, using the primitive rules above, see Subsection 2.1, or some similar system of formula transformations) we can propose the Newton formula

$$\sum_{j=0}^{k} (-1)^j \sigma_j \tau_{k-j} = 0$$

where $\tau_l = \sum_{i=1}^{n} x_i^l$ and σ_j are elementary symmetric functions for which there are depth three formulas (over zero characteric fields) obtained from the Lagrange interpolation polynomial due to M. Ben-Or (see, e.g., [Shp00]).

Finally, we ask whether using other kinds of formulas can make $\mathcal{F}$-$\mathcal{PC}$ or $\mathcal{F}$-$\mathcal{NS}$ stronger, e.g., one may try using the exponentiation which would allow to use F^d for exponentially large d in a polynomial-size proof.

9 Acknowledgements

The second author would like to thank Boris Konev for useful discussions and V. P. Orevkov for providing multiple references.

References

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6:83–96, 1986.

[AR00] M. Alekhnovich and A. A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. Manuscript, `http://genesis.mi.ras.ru/~razborov/misha.ps`, September 2000.

[BCR98] J. Bochnak, M. Coste, and M.-F. Roy. *Real algebraic geometry.* Springer-Verlag, 1998.

[BGIP99] S. Buss, D. Grigoriev, R. Impagliazzo, and T. Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing, STOC'99*, pages 547–556, 1999. A journal version is to appear in Journal of Computer and System Sciences.

[BIK+96] P. Beame, R. Impagliazzo, J. Krajíček, T. Pitassi, and P. Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc.*, 73(3):1–26, 1996.

[BIK+97] P. Beame, 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):256–298, 1996/97.

[Bus87] S. Buss. Polynomial size proofs of the propositional pigeonhole principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[CEI96] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing, STOC'96*, pages 174–183, 1996.

[CR79] S. A. Cook and A. R. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[GR00] D. Grigoriev and A. Razborov. Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Applicable Algebra in Engineering, Communication and Computing*, 10(6):465–487, 2000.

[Gri99] D. Grigoriev. Linear lower bound on degrees of Positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 1999. To appear.

[Gri00] D. Grigoriev. Complexity of Positivstellensatz proofs for the knapsack. *Computational Complexity*, 2000. To appear.

[GV01] D. Grigoriev and N. Vorobjov. Complexity of Null- and Positivstellensatz proofs. *Annals of Pure and Applied Logic*, 2001. To appear.

[IPS99] R. Impagliazzo, P. Pudlák, and J. Sgall. Lower bounds for the polynomial calculus. *Computational Complexity*, 8(2):127–144, 1999.

[KPW95] J. Krajíček, P. Pudlák, and A. Woods. Exponential lower bound to the size of bounded depth frege proofs of the pigeonhole principle. *Random Structures and Algorithms*, 7:15–39, 1995.

[Kra95] J. Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1995.

[LS91] L. Lovász and A. Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM Journal on Optimization*, 1:166–190, 1991.

[PBI93] T. Pitassi, P. Beame, and R. Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Computational Complexity*, 3:97–140, 1993.

[Pit97] T. Pitassi. Algebraic propositional proof systems. In Neil Immerman and Phokion G. Kolaitis, editors, *Descriptive Complexity and Finite Models*, volume 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1997.

[Pud99] P. Pudlák. On the complexity of propositional calculus. In *Sets and Proofs: Invited papers from Logic Colloquium'97*, pages 197–218. Cambridge University Press, 1999.

[Raz98] A. A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7:291–324, 1998.

[Shp00] A. Shpilka. Affine projections of symmetric polynomials. Submitted, 2000.

[vdW31] B. L. van der Waerden. *Moderne Algebra*. Springer–Verlag, 1st edition, 1930/31.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'